

スーパーコンピューティング ニュース

Vol.24 No.4, 2022.7



東京大学情報基盤センター
INFORMATION TECHNOLOGY CENTER, THE UNIVERSITY OF TOKYO

スーパーコンピュータシステム 利用負担金表

Wisteria/BDEC-01 スーパーコンピュータシステム 利用負担金表(2021 年 5 月 14 日)

区分	負担金額(税込)		ディスク容量	備考
	大学・公共機関等	企業		
一般申込 (基本セット) Wisteria-O/A	申込 1 セット当り 60,000 円 (8,640 トークン)		申込 1 セット当り /work 2TB 利用者当り /home 50GB	利用期間 12 か月の金額・トークン量 (利用期間は1ヶ月単位で設定可)
公募制度による申込 Wisteria-O	申込 1 セット当り 60,000 円 (8,640 トークン)	申込 1 セット当り 72,000 円 (8,640 トークン)	申込 1 セット当り /work 2TB 利用者当り /home 50GB	
公募制度による申込 Wisteria-A	申込 1 セット当り 180,000 円 (25,920 トークン)	申込 1 セット当り 216,000 円 (25,920 トークン)	申込 1 セット当り /work 6TB 利用者当り /home 50GB	
GPU 専有申込 (公募制度の申込可) Wisteria-A	申込 1GPU セット当り 270,000 円 (25,920 トークン)	申込 1GPU セット当り 324,000 円 (25,920 トークン)	申込 1GPU セット当り /work 6TB 利用者当り /home 50GB	利用期間 12 か月の金額・トークン量 (利用期間は1ヶ月単位で設定可) 1, 2, 4GPU のみ申込可, 申込単位は下表参照
ノード固定 (公募制度の申込可) Wisteria-A	2,160,000 円 (207,360 トークン)	2,592,000 円 (207,360 トークン)	/work 48TB 利用者当り /home 50GB	利用期間 12 か月の金額・トークン量 (利用期間は1ヶ月単位で設定可) 1 セットのみ申込可
一般申込 (最小セット) Wisteria-O/A	5,000 円 (720 トークン)		/work 2 TB 利用者当り /home 50 GB	利用期間は当該年度末まで
トークン量追加	5,000 円 (720 トークン)	6,000 円 (720 トークン)		
ディスク容量追加	6,480 円／(1TB*年)			1TB 単位で申込可 (/work のみ)

※Wisteria/BDEC-01 においてはパーソナルコースとグループコースの区分を廃止し、これまでのパーソナルコースは一般申込に統合した。

※Wisteria-O のトークン消費係数は 1.00 (1 ノード当り) , Wisteria-A のトークン消費係数は 3.00 (1GPU 当り)である。

Wisteria-O にトークン消費係数 1.50 のノード群(優先利用向け)を全体の 15%程度設ける。

※Wisteria-O の 1 ジョブで利用可能な最大ノード数は 2,304 ノード, Wisteria-A の 1 ジョブで利用可能な最大 GPU 数は 64GPU

※括弧()内は付与するトークン量。実行したジョブのノード時間積または GPU 時間積と消費係数に応じてトークンが消費される。

付与したトークンは、利用期間内に全量が使用できることを保証するものではない。

トークンは利用期間内に限り有効とし、利用終了後に残量がある場合でも繰越や利用負担金の返還は行わない。

トークンの他のシステムへの移行については、「トークン移行におけるトークン量の換算表」を参照。

※公募制度による申し込み、ノード固定の申し込みには審査を要する。

※/home のディスク容量は複数のグループに所属している場合でも利用者当り 50GB 固定。

※GPU 専有申込の申込単位

GPU 数	トークン量	大学・公共機関等	企業
1	25,920	270,000 円	324,000 円
2	51,840	540,000 円	648,000 円
4	103,680	1,080,000 円	1,296,000 円

Oakbridge-CX スーパーコンピュータシステム 利用負担金表(2022 年 4 月 1 日)

区分	負担金額(税込)		ディスク容量	備考
	大学・公共機関等	企業		
一般申込	申込 1 セット当り 100,000 円 (8,640 トークン)	申込 1 セット当り 120,000 円 (8,640 トークン)	グループ 1 セット当り /work 4TB 利用者当り /home 50GB	利用期間 12 か月の金額・トークン量 (利用期間は1ヶ月単位で設定可)
ノード固定	申込 1 セット当り 150,000 円 (8,640 トークン)	申込 1 セット当り 180,000 円 (8,640 トークン)		
一般申込 (最小セット)	8,400 円 (720 トークン)		/work 4 TB 利用者当り /home 50 GB	利用期間は当該年度末まで
トークン量追加	8,400 円 (720 トークン)	10,000 円 (720 トークン)		
ディスク容量追加	6,480 円/(1TB*年)			1TB 単位で申込可 (/work のみ)

※Oakbridge-CX においてもパーソナルコースとグループコースの区分を廃止し、これまでのパーソナルコースは一般申込に統合した。

※トークン消費係数は 1.00、ただしトークン消費係数 1.50 のノード群(優先利用向け)を全体の 15%程度設ける。

※1 ジョブで利用可能な最大ノード数は 256 ノード。

※括弧()内は付与するトークン量。実行したジョブのノード時間積と消費係数に応じてトークンが消費される。

付与したトークンは、利用期間内に全量が使用できることを保証するものではない。

トークンは利用期間内に限り有効とし、利用終了後に残量がある場合でも繰越や利用負担金の返還は行わない。

トークンの他のシステムへの移行については、「トークン移行におけるトークン量の換算表」を参照。

※ノード固定の申し込みには審査を要する。

※/home のディスク容量は複数のグループに所属している場合でも利用者当り 50GB 固定。

トークン移行におけるトークン量の換算表

移行先	Oakbridge-CX システム	Wisteria/BDEC-01 システム
移行元		
Oakbridge-CX システム	—	1.6
Wisteria/BDEC-01 システム	0.6	—

移行先に追加されるトークン量＝移行トークン量×係数

注意事項(Wisteria/BDEC-01, Oakbridge-CX スーパーコンピュータシステム 共通)

- ・「大学・公共機関等」は大学、高等専門学校及び大学共同利用機関、文部科学省所管の独立行政法人、学術研究及び学術振興を目的とする国又は地方公共団体が所管する機関、並びに文部科学省科学研究費補助金の交付を受けて学術研究を行う者に適用する。
- ・「企業」の申し込みには、企業利用申込書添付書類の提出および審査を要する。
- ・利用期間は、利用開始月から終了月の末日またはサービス休止前までとする。利用期間内に計算機利用を中止した場合であっても利用負担金額の変更は行わない。年度の途中で利用開始または終了する場合の負担金額は月数別利用負担金表(Web ページ)を参照すること。
- ・前掲の利用負担金表は基本セットの内容であり、最小セットについては Web ページを参照すること。
 - － 利用開始月が 4 月から 9 月までは 12 月、10 月から 12 月までは 3 月、1 月から 3 月までは 3 月末。
 - － 前年度内に事前申込をした分については、利用開始月に関わらず、11 月の請求となる。
- ・利用負担金額が減額となる変更はできない。
- ・ディスク量は、グループ全体の上限值である。

スーパーコンピュータシステム ジョブクラス制限値

Wisteria/BDEC-01 スーパーコンピュータシステム (Wisteria-O) ジョブクラス制限値 (2021 年 5 月 14 日)

キュー名※1	ノード数※2 (最大コア数)	制限時間 (経過時間)	メモリー容量 (GiB)※3	一般申込	公募制度 による申込
debug-o	1 ~ 144 (6,912)	30 分	28	○	○
short-o	1 ~ 72 (3,456)	8 時間	28	○	○
(regular-o)					
small-o	1 ~ 144 (6,912)	48 時間	28	○	○
medium-o	145 ~ 576 (27,648)	"	"	○	○
large-o	577 ~ 1,152 (55,296)	"	"	○	○
x-large-o	1,153 ~ 2,304 (110,592)	24 時間	"	○	○
priority-o	1 ~ 288 (13,824)	48 時間	28	○	○
challenge-o	1 ~ 7,680 (368,640)	24 時間	28	★	★
(interactive-o) ※4					
interactive-o_n1	1 (48)	30 分	28	○	○
interactive-o_n12	2 ~ 12 (576)	10 分	"	○	○
prepost	1 (56)	6 時間	340	○	○
prepost1_n1 ~ prepost4_n1	1 (56)	1~6 時間	340	○	○
prepost1_n4	1 ~ 4 (224)	1~6 時間	340	○	○
prepost1_n8	1 ~ 8 (448)	1~6 時間	340	○	○

★ 審査による課題選定の上、月1回の一定期間のみ利用可能(原則として月末処理日前日の朝~翌日朝)

※1 キューの指定("#PJM -L "rscgrp=キュー名" ") は、regular-o、debug-o、short-o を小文字で指定する
regular-o キューはノード数の指定("#PJM -L "node=ノード数" ") でノード数別のキューに投入される

※2 トークンの消費係数は1ノード当り1.00。ただし priority-o は優先利用ノード群のためトークン消費係数は1.50

※3 1ノード当りの利用者が利用可能なメモリー容量

※4 インタラクティブジョブの起動は次のとおり (トークン消費なし)

pjsub --interact -g グループ名 -L "rscgrp=interactive-o,node=ノード数"

Wisteria/BDEC-01 スーパーコンピュータシステム (Wisteria-A) ジョブクラス制限値 (2021 年 5 月 14 日)

キュー名※1	ノード数・GPU 数※2 (最大 GPU 数)	制限時間 (経過時間)	メモリー 容量 (GiB) ※3	一般申込	公募制度 による申込	GPU 専有申込	ノード固定
debug-a	1 ノード (8)	30 分	448	○	○	○	○
short-a	1 ~ 2 ノード (16)	2 時間	448	○	○	○	○
(regular-a)							
small-a	1 ~ 2 ノード (16)	48 時間	448	○	○	○	○
medium-a	3 ~ 4 ノード (32)	"	"	○	○	○	○
large-a	5 ~ 8 ノード (64)	24 時間	"	○	○	○	○
share-debug	1, 2, 4 GPU	30 分	56	○	○	○	○
share-short	1, 2, 4 GPU	2 時間	56	○	○	○	○
(share)							
share-1	1 GPU	48 時間	56	○	○	○	○
share-2	2 GPU	"	"	○	○	○	○
share-4	4 GPU	24 時間	"	○	○	○	○
challenge-a	1 ~ 39 ノード (312)	24 時間	448	★	★	★	★
任意	1 ノード (8)	任意 ※4	448	×	×	○	○
interactive-a ※5	1 ノード (8)	10 分	56	○	○	○	○
share-interactive	1 GPU	"	"	○	○	○	○

★ 審査による課題選定の上、月1回の一定期間のみ利用可能(原則として月末処理日前日の朝~翌日朝)

※1 キューの指定("#PJM -L "rscgrp=キュー名" ") は、regular-a、debug-a、short-a を小文字で指定する
regular-a キューはノード数の指定("#PJM -L "node=ノード数" ") でノード数別のキューに投入される

※2 トークンの消費係数は1GPU 当り3.00

※3 1ノード当りの利用者が利用可能なメモリー容量

※4 申込ノード数の合計以内ならば、キュー名・制限時間(原則 48 時間以内)は相談の上、任意に設定可能

※5 インタラクティブジョブの起動は次のとおり (トークン消費なし)

pjsub --interact -g グループ名 -L "rscgrp=interactive-a,node=ノード数"

Oakbridge-CX スーパーコンピュータシステム ジョブクラス制限値(2019 年 7 月 1 日)

キュー名※1	ノード数※2 (最大コア数)	制限時間 (経過時間)	メモリー 容量 (GB) ※3	一般 申込	ノード 固定
debug	1 ~ 16 (896)	30 分	168	○	○
short	1 ~ 8 (448)	8 時間	168	○	○
(regular)					
small	1 ~ 16 (896)	48 時間	168	○	○
medium	17 ~ 64 (3584)	"	"	○	○
large	65 ~ 128 (7168)	"	"	○	○
x-large	129 ~ 256 (14336)	24 時間	"	○	○
challenge	1 ~ 1368 (76608)	24 時間	168	★	★
任意	申込数	任意 ※4	168	×	○
(interactive) ※5					
interactive_n1	1 (56)	2 時間	168	○	○
interactive_n8	2 ~ 8 (448)	10 分	"	○	○

★ 審査による課題選定の上、月1回の一定期間のみ利用可能(原則として月末処理日前日の朝～翌日朝)

※1 キューの指定(“#PJM -L “rscgrp=キュー名””)は、regular, debug, short を小文字で指定する

regular キューはノード数の指定(“#PJM -L “node=ノード数””)でノード数別のキューに投入される

※2 トークンの消費係数は 1.00

※3 1ノード当りの利用者が利用可能なメモリー容量

※4 申込ノード数の合計以内ならば、キュー名・制限時間(原則 48 時間以内)は相談の上、任意に設定可能

※5 インタラクティブジョブの起動は次のとおり (トークン消費なし)

pjsub --interact -g グループ名 -L “rscgrp=interactive,node=ノード数”

大規模共通ストレージシステム 利用負担金表

大規模共通ストレージシステム(第1世代、Ipomoea-01) 利用負担金表(2022 年 6 月 1 日)

区分	負担金額(税込)
一般申込	【大学・公共機関等 7,200 円、企業 8,640 円】(1TB の場合、年額) (ディスク容量ごとの負担金額は下表参照、利用期間は 1 ヶ月単位で設定可) 利用者番号登録数 制限なし
	並列ファイルシステム ディスク容量
	1 TB
	[10 TB まで 1 TB 追加当たり]
	10 TB
	[100 TB まで 1 TB 追加当たり]
	100 TB
	[1,000 TB まで 1 TB 追加当たり]
	1,000 TB
	[以降 1 TB 追加当たり]
ディスク容量追加	大学・公共機関等
	企業
	7,200 円/年
	8,640 円/年
	4,200 円/年
	5,040 円/年
	45,000 円/年
	54,000 円/年
	3,000 円/年
	3,600 円/年
ディスク容量追加	315,000 円/年
	378,000 円/年
	[2,400 円/年]
	[2,880 円/年]
	2,475,000 円/年
	2,970,000 円/年
	[2,100 円/年]
	[2,520 円/年]
	大学・公共機関等
	企業
ディスク容量追加	7,200 円/年
	8,640 円/年
	4,200 円/年
	5,040 円/年
	3,000 円/年
	3,600 円/年
	2,400 円/年
	2,880 円/年
	2,100 円/年
	2,520 円/年

※利用期間については利用開始月から当該年度のサービス終了月までとし、年度を超えないものとする。利用期間の指定がある場合は利用終了月までとする。

※ディスク容量は利用期間内に限り有効とし、利用終了後に残存しているデータは削除するものとする。

※ディスク容量追加の負担金額は追加単位額に追加する資源量および利用期間を乗じたものとする。

※ファイル、ディレクトリの総数制限についてはディスク容量に比例した値を別途定めるものとする。

センターから

サービス休止等のお知らせ

2022 年 7 月下旬からの計算機及びストレージシステムのサービス予定は以下のとおりです。

Wisteria/BDEC-01 スーパーコンピュータシステム

○ Wisteria/BDEC-01 スーパーコンピュータシステム サービス休止のお知らせ

日 付	利用者サービス	センター内作業
7 月 29 日 (金)	9:00 ～ 22:00 までサービス休止	月末処理
8 月 19 日 (金) ～ 8 月 25 日 (木)	8/19(金)9:00 ～ 8/25(木)17:00 までサービス休止	柏 II キャンパスにおける特別高圧受変電 設備点検・二次変電設備点検、月末処理
9 月 16 日 (金)	9:00 ～ 22:00 までサービス休止	月末処理

- ・ Wisteria/BDEC-01 システムは、原則 24 時間サービスを行っています。
ただし、月末処理日（原則として毎月最終金曜日）はサービスを停止します。

○ Wisteria/BDEC-01 スーパーコンピュータシステム 大規模 HPC チャレンジのお知らせ (*)

大規模 HPC チャレンジ 実施期間
7 月 28 日 (木) 9:00 ～ 29 日 (金) 9:00 まで 8 月 18 日 (木) 9:00 ～ 19 日 (金) 9:00 まで 9 月 15 日 (木) 9:00 ～ 16 日 (金) 9:00 まで

- ・ 上記期間中、Wisteria/BDEC-01 の debug-o/a, short-o/a, regular-o/a, priority-o, interactive-o/a, prepost, share, share-debug, share-short, share-interactive, ノード固定及び講義用キューのサービスを休止します。
ログインノードは通常どおり利用できます。

Oakbridge-CX スーパーコンピュータシステム

○ Oakbridge-CX スーパーコンピュータシステム サービス休止のお知らせ

日 付	利用者サービス	センター内作業
7 月 27 日 (水)	9:00 ～ 20:00 までサービス休止	月末処理
8 月 8 日 (月) ～ 8 月 15 日 (月)	8/8(月) 9:00 ～ 8/15(月) 17:00 までサービス休止	電力消費節減のためサービス休止
8 月 26 日 (金)	9:00 ～ 20:00 までサービス休止	月末処理
9 月 22 日 (木) ～ 9 月 26 日 (月)	9/22(木) 9:00 ～ 9/26(月) 17:00 までサービス休止	月末処理、空調機メンテナンス、柏キャンパスにおける特別高圧受変電設備点検・二次変電設備点検

- ・ Oakbridge-CX システムは、原則 24 時間サービスを行っています。
ただし、月末処理日（原則として毎月最終水曜日）はサービスを停止します。

○ Oakbridge-CX スーパーコンピュータシステム 大規模 HPC チャレンジのお知らせ (*)

大規模 HPC チャレンジ 実施期間
7 月 26 日 (火) 9:00 ～ 27 日 (水) 9:00 まで 8 月 25 日 (木) 9:00 ～ 26 日 (金) 9:00 まで 9 月 21 日 (水) 9:00 ～ 22 日 (木) 9:00 まで

- ・ 上記期間中、Oakbridge-CX の debug, short, regular, interactive, prepost, ノード固定 及び 講義用キューのサービスを休止します。ログインノードは通常どおり利用できます。

大規模共通ストレージシステム (Ipomoea-01)

○ Ipomoea-01 サービス休止のお知らせ

日 付	利用者サービス	センター内作業
9 月 22 日 (木) ～ 9 月 26 日 (月)	9/22(木) 9:00 ～ 9/26(月) 17:00 までサービス休止	月末処理、空調機メンテナンス、柏キャンパスにおける特別高圧受変電設備点検・二次変電設備点検

- ・ Ipomoea-01 は、原則 24 時間サービスを行っています。
ただし、月末処理実施のためサービスを停止する場合があります。

【注意事項】

- ・ サービス休止等の計画は原稿作成時の予定です。やむを得ずサービスを変更したり、休止したりする場合がありますので、最新の情報は login 時のメッセージ及びスーパーコンピューティング部門の Web ページの運用スケジュール (<https://www.cc.u-tokyo.ac.jp/supercomputer/schedule.php>) をご確認ください。
- ・ 平日の9:00～17:00以外、休日(土・日・祝日等)は、システム障害等でサービスが停止した場合、運転を継続できない場合があります。その場合は、その時間をもってサービスを中止しますのでご了承ください。
- * Wisteria/BDEC-01 及びOakbridge-CX における大規模HPC チャレンジについて、新型コロナウイルス感染症の状況次第で実施時間・実施条件の変更や、中止となる可能性があります。詳細はWeb ページ(<https://www.cc.u-tokyo.ac.jp/guide/hpc/>)をご覧ください。

システム変更等のお知らせ

(2022.5.1 - 2022.6.30 変更)

1. ハードウェア

- 1.1 Wisteria/BDEC-01 スーパーコンピュータシステム … なし
- 1.2 Oakbridge-CX スーパーコンピュータシステム … なし
- 1.3 Ipomoea-01 大規模共通ストレージシステム … サービス開始 (2022.6.1)

全体構成

	項目	機器諸元
並列ファイルシステム	ファイルシステム	DDN EXAScaler (Lustre ベース)
	ストレージ容量	25.9 PB
	i-node 数上限	168 億
	ストレージデータ転送速度	125 GB/s
	MDS + MDT	DDN 1U サーバ × 4 台 + DDN SFA200NV × 1 台
	メタデータ格納デバイス	NVMe SSD 3.84 TB × 17
	OSS + OST	DDN ES7990X × 5 台 + SS9012 × 20 台
	ファイルデータ格納デバイス	SAS-HDD 18 TB × 383 (OSS × 1 台 + OST × 4 台 あたり)

2. ソフトウェア

2.1 Red Hat Enterprise Linux 8 (Wisteria/BDEC-01)

➤ Odyssey

WaitIO	v1.0	(2022.5.27)
HΦ	3.5.0	(2022.5.27)
GENESIS	2.0beta3	
MODYLAS	1.1.0β	
NTChem	12.1.1	
OpenMX	3.9.9	
PHASE/0	2021.01	
SALMON	2.0.2	
SMASH	3.0.0	
FrontISTR	5.2	
mVMC	1.2.0	
AkaiKKR	Cpa2002v010	
ALAMODE	1.3.0	
Phonopy	2.11.0	

➤ Aquarius

MATLAB	R2022a	(2022.5.27)
NVIDIA HPC SDK	v22.2、v22.5	(2022.5.27)
gfarm	2.7.20	(2022.6.24)
gfarm2fs	1.2.16	

➤ ログインノード

gfarm	2.7.20	(2022.6.24)
gfarm2fs	1.2.16	

インストールを実施しました。利用方法については、利用支援ポータルのお知らせ、またはドキュメント閲覧より利用手引書をご覧ください。

2.2 Red Hat Enterprise Linux 7, CentOS 7 (Oakbridge-CX)

SingularityCE	v3.9.5	(2022.5.25)
MATLAB	R2022a	(2022.5.25)
AkaiKKR	cpa2002v010	(2022.5.25)
ALAMODE	1.3.0	
FrontISTR	5.2	
GENESIS (倍精度、単精度)	1.7.0	
HΦ	3.5.0	
mVMC	1.2.0	
NTChem	12.1.1	
PHASE/0 (2D、3D)	2021.01	
Phonopy	2.12.0	
SALMON	2.0.2	
SMASH	3.0.0	
gfarm	2.7.20	(2022.6.22)
gfarm2fs	1.2.16	

インストールを実施しました。利用方法については、利用支援ポータルのお知らせ、またはドキュメント閲覧より利用手引書をご覧ください。

2.3 Red Hat Enterprise Linux 8 (Ipomoea-01) … サービス開始 (2022.6.1)

※ ログインノード上のソフトウェアは主にファイル転送操作や転送用のツール等で利用

項目	構成
OS	Red Hat Enterprise Linux 8
コンパイラ	GNU コンパイラ
メッセージ通信ライブラリ	Open MPI
開発環境	OpenJDK
フリーソフトウェア	autoconf、automake、bash、bzip2、cvs、emacs、findutils、gawk、gdb、make、grep、gnuplot、gzip、less、m4、python2、python3、perl、ruby、screen、sed、subversion、tar、tcsh、tcl、vim、zsh、git など

3. その他

3.1 Ipomoea-01 サービス開始に伴う Wisteria/BDEC-01 及び Oakbridge-CX ログインノード群におけるファイルシステムマウントについて

Ipomoea-01 大規模共通ストレージシステムは 2022 年 6 月 1 日からサービスを開始しました。Wisteria/BDEC-01 のログインノード群においては 2022 年 6 月 1 日から、Oakbridge-CX のログインノード群においては 2022 年 6 月 22 日から Ipomoea-01 ファイルシステム上のファイルの読み込み、書き込みが可能になっています。本誌「大規模共通ストレージシステム(第 1 世代) 運用に関するお知らせ(再掲)」、当センター Web ページや Ipomoea-01 利用支援ポータルもご参照ください。

Wisteria/BDEC-01 及び Oakbridge-CX ログインノード群における Ipomoea-01 ファイルシステムへのアクセスパスは以下のとおりとなります。

- /common1/home/ユーザ名(ユーザ ID) : 利用者ごとの領域
- /common1/work/グループ名(プロジェクトコード) : グループごとの領域

3.2 MATLAB R2022a のインストールについて

2022 年 2 月 25 日より提供を開始 (<https://www.cc.u-tokyo.ac.jp/guide/application/introduction-matlab.php>)した MATLAB R2021b に加えて、MATLAB R2022a を Wisteria/BDEC-01 Aquarius 及び Oakbridge-CX にインストールしました。当センターでは、MATLAB におけるスーパーコンピュータ利用者の自作あるいはオープンソースのプログラムから関数として呼び出す機能が重要と考えています。特にデータ解析、機械学習系に関連する豊富な機能は、利用者の計算科学シミュレーションコードを高度化し、研究の幅を広げることに貢献するものと期待しております。MATLAB については本誌記事「Oakbridge-CX と Wisteria/BDEC-01 で稼働する技術計算言語 MATLAB のご紹介(MathWorks Japan)」をご覧くださいと共に、本誌記事「第 177 回お試しアカウント付き並列プログラミング講習会「MATLAB の実行方法」実施報告」もご参考ください。

大規模共通ストレージシステム(第1世代) 運用に関するお知らせ (再掲)

スーパーコンピューティングチーム

東京大学情報基盤センター(以下「当センター」)におきまして、当センターで運用する各スーパーコンピュータシステムからアクセス可能なストレージシステム：大規模共通ストレージシステム(第1世代) (Ipomoea-01)を 2022 年 6 月からサービスを開始しました。ここでは、ご利用に際して必要な手続き及び負担金をお知らせいたします。

1. 利用申込みについて

ご利用になるための利用申込み方法の詳細は当センター Web ページもご参照ください。Ipomoea-01 をご利用になる場合には、当センターのスーパーコンピュータシステム Oakbridge-CX、Wisteria/BDEC-01 をすでにご利用かどうかにより、Ipomoea-01 の利用申込み手続きが必要なケースがあります。なお、Ipomoea-01 におけるディスク容量追加手続きは可能ですが、スーパーコンピュータシステムにユーザ ID を有するかに係わらず、容量追加のための手続きが別途必要になります。

1.1 2021 年度末に Oakforest-PACS にユーザ ID を有していなかった利用者、または、2021 年度末に Oakforest-PACS にユーザ ID を有していたが、Oakforest-PACS のファイル移行サービスを希望しなかった利用者

- 2022 年度 Oakbridge-CX、Wisteria/BDEC-01 にユーザ ID を有する場合
利用申し込み手続きは不要です。ご利用のユーザ ID で Ipomoea-01 を利用できます(教育利用、講習会を除く)。
 - 利用者ごとの領域に 5 TB の無償分のディスク容量を付与します。
 - 登録されているスーパーコンピュータシステムで付与されているグループのディスク容量の 15 %を各プロジェクトコードのグループごとの領域として無償で付与します。(トークン移行先のシステムを除く)
- 2022 年度 Oakbridge-CX、Wisteria/BDEC-01 にユーザ ID を持たない場合
利用申し込み手続きが必要です。無償分のディスク容量は設定されません。

1.2 2021 年度末に Oakforest-PACS にユーザ ID を有しており、ファイル移行サービスを希望した利用者

- 2022 年度 Oakbridge-CX、Wisteria/BDEC-01 にユーザ ID を有する場合
利用申し込み手続きは不要です。ご利用のユーザ ID で Ipomoea-01 を利用できます(教育利用、講習会を除く)。
 - 利用者ごとの領域に 5 TB の無償分のディスク容量を付与します。
 - 移行を希望したグループごとの領域に 5 TB の無償分のディスク容量を付与します。ただし、トークン移行により Oakbridge-CX、Wisteria/BDEC-01 のどちらかに同プロジェクトコードのグループが存在する場合は、グループごとの領域への付与ディスク容量は、登録されているスーパーコンピュータシステムで付与されているグループのディスク容量の 15% とします。
 - 登録されているスーパーコンピュータシステムで付与されているグループのディスク容量の 15 % を各プロジェクトコードのグループごとの領域として無償で付与します。(トークン移行先のシステムを除く)
- 2022 年度 Oakbridge-CX、Wisteria/BDEC-01 にユーザ ID を持たない場合
利用申し込み手続きは不要で、Oakforest-PACS のユーザ ID で 2022 年 11 月 30 日までは無償でご利用いただけます。2022 年 12 月 1 日以降も移行したファイルを利用するためには有償の利用申し込み手続きが必要です。
 - 利用者ごと、移行を希望したグループごとの領域に 5 TB の無償分のディスク容量を付与します。
 - 2022 年 11 月までに継続利用の申し込みがない場合、2022 年 12 月以降にファイルは削除されます。

2. 利用負担金について

Ipomoea-01 をご利用になる場合の利用負担金は、運用を開始した 2022 年 6 月以降発生します。

2.1 2021 年度末に Oakforest-PACS にユーザ ID を有していなかった利用者、または、2021 年度末に Oakforest-PACS にユーザ ID を有していたが、Oakforest-PACS のファイル移行サービスを希望しなかった利用者

- 2022 年度 Oakbridge-CX、Wisteria/BDEC-01 にユーザ ID を有する場合
無償分のディスク容量の範囲でご利用の場合、負担金は発生しません。別途 Ipomoea-01 でのディスク容量の追加手続きを行った場合に、**追加分の利用負担金が発生します。**
- 2022 年度 Oakbridge-CX、Wisteria/BDEC-01 にユーザ ID を持たない場合
利用申し込みディスク容量に対する負担金が発生します。

2.2 2021 年度末に Oakforest-PACS にユーザ ID を有しており、ファイル移行サービスを希望した利用者

- 2022 年度 Oakbridge-CX、Wisteria/BDEC-01 にユーザ ID を有する場合
利用者ごとの領域については、無償分のディスク容量の範囲でご利用の場合、負担金は発生しません。移行を希望したグループごとの領域については 2022 年 11 月まで負担金は発生しません。**2022 年 12 月以降も移行を希望したグループごとの領域を利用するためには有償の利用申し込み手続きが必要です。**別途 Ipomoea-01 でのディスク容量の追加手続きを行った場合に、**追加分の利用負担金が発生します。**
- 2022 年度 Oakbridge-CX、Wisteria/BDEC-01 にユーザ ID を持たない場合
2022 年 11 月まで負担金は発生しません。2022 年 12 月以降は**利用申し込みディスク容量に対する負担金が発生します。**

3. 運用開始後のサービスについて

3.1 ログインノードサービス

データ転送やそのためのツールを利用するための環境として、ログインノードを 2 台用意しています。ログインノードは、当センターでサービスを行っている他のスーパーコンピュータシステムと同様に、公開鍵認証方式による接続となります。鍵登録の方法や、接続ホスト名などの詳細については、当センター Web ページや Ipomoea-01 利用支援ポータル参照してください。

3.2 システムの利用イメージ

Ipomoea-01 のご利用イメージは、おおよそ図 1 の通りです。

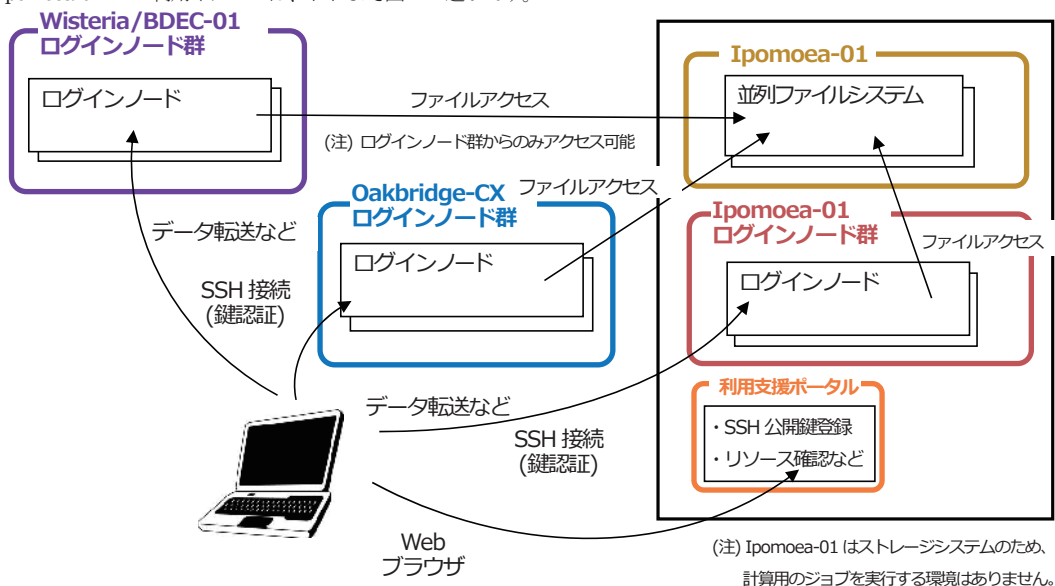


図 1. システムのご利用イメージ

3.3 ファイルシステム

ファイルシステムは、システム構成図 (図 1) にある通り、並列ファイルシステムのみで構成されます。

並列ファイルシステムは DDN 社製の Lustre ベースの DDN EXAScaler で提供されます。主に以下の 2 つのディレクトリにファイルを保存することができ、Ipomoea-01 のログインノード群からだけではなく、Oakbridge-CX、Wisteria/BDEC-01 のログインノード群からもファイルアクセス (ファイルの読み込み、書き込み) が可能です。その他、詳細については当センター Web ページや Ipomoea-01 利用支援ポータルにてお知らせいたします。

- `/home/ユーザ名(ユーザ ID)`: 利用者ごとの領域
- `/work/グループ名(プロジェクトコード)`: グループごとの領域

4. その他

最新の情報は、当センター Web ページ (<https://www.cc.u-tokyo.ac.jp>) にて随時ご案内いたします。メールによる問い合わせについては、事前に Web ページで情報がないかご確認の上、受付窓口 uketsuke@cc.u-tokyo.ac.jp までお願いいたします。

Oakbridge-CX と Wisteria/BDEC-01 で稼働する

技術計算言語 MATLAB のご紹介

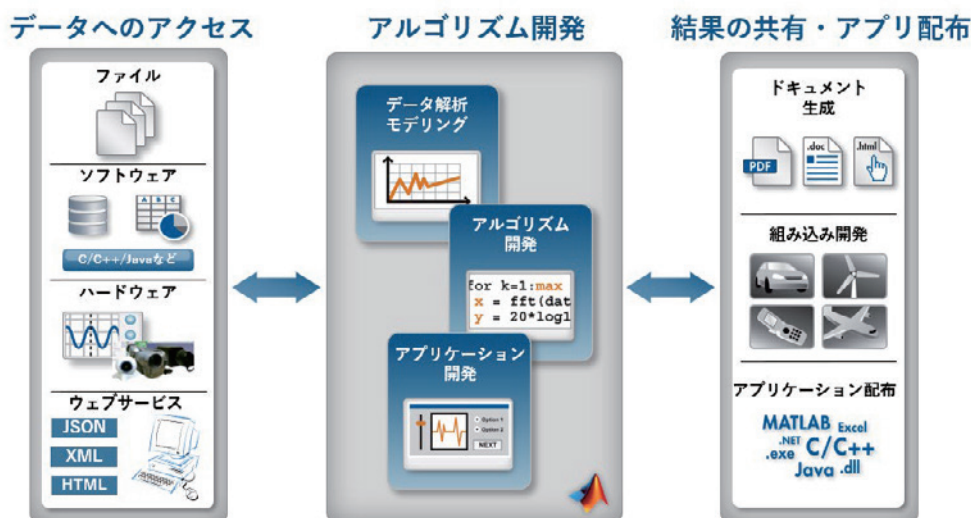
齊藤 甲次朗¹, 沖田 芳雄²

MathWorks Japan

1. はじめに

MATLAB[®]は、数百万人のエンジニアや科学者が、データの解析やアルゴリズムの開発、モデルの作成に使用しているプログラミングおよび数値計算プラットフォームです。1970 年代後期に、Fortran で作成された会話型の行列計算電卓ソフトとして誕生しました (Classic MATLAB)。基となった LINPACK と EISPACK は、当時の行列計算における最先端の Fortran サブルーチン ライブラリでしたが、学生達が Fortran プログラムを書かずに、簡単に使うことができるように、との目的で開発されたものです。パーソナルコンピュータの黎明期を通じて、行列計算に不可欠な浮動小数点演算が可能となったことから、もっと強力で生産性の高い技術計算環境がエンジニアや科学者には必要である、との認識の元、1984 年に MathWorks 社が創設され、今の MATLAB が誕生しました³。

30 年以上に渡る開発を通じて、MATLAB は強力なグラフィックス機能を備え、各種データへのアクセス、既存ライブラリの活用 (C/C++/Fortran/Python[®]) から、アルゴリズム開発、C/C++/GPU コード生成、組み込み実装まで一貫した開発を可能とする、単なる技術計算言語の枠を超えた統合開発環境へと成長しました。



第 1 図: 統合開発環境としての MATLAB

¹ シニア アプリケーションエンジニア

² シニア カスタマーサクセスエンジニア

³ A History of MATLAB, Proceedings of the ACM on Programming Languages, Vol. 4, No. HOPL, June 2020

この間、様々なユーザーニーズに答えることで、制御設計、信号処理・通信、画像処理、テストや計測、情報生命科学、金融工学におけるモデリングや解析、統計・機械学習／深層学習、自動運転支援などの専門分野向けの幅広いアドオン製品を開発してきました。東京大学でも導入いただいた全学利用ライセンス（Campus-Wide License）では、スーパーコンピューターシステムで稼働する MATLAB Parallel Server™も含め、学生や教職員は、100 種類以上に渡る、ほぼすべての MathWorks 製品を無制限で利用可能です。またこのスーパーコンピューターシステムを利用可能なアカデミックユーザーは、東京大学のライセンスを利用することができます。

2. MATLAB と他のプログラミング言語との連携

MATLAB エンジン API を使用すると、MATLAB のデスクトップセッションを開始せずに、他のプログラミング言語内から MATLAB コマンドを実行することが可能になります。MATLAB エンジン API は、以下の言語から利用可能⁴で、スーパーコンピューターシステムからも、豊富なアドオン製品群をご利用いただけます。

- C/C++
- Fortran
- Java®
- Python



また別のプログラミング言語で開発されたコードを MATLAB から呼び出すことができます⁵。このため、過去に開発された資産を無駄にすることなく、また異なるプログラミング言語を使用するチームが別の言語で再コーディングするための時間を減らしながら、研究開発にかかる時間を増やすことができます。



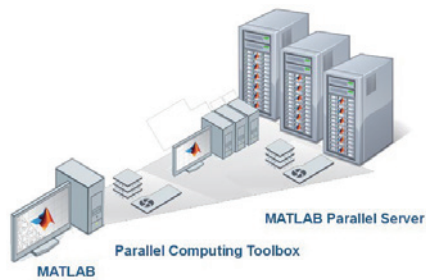
3. MATLAB の並列処理の概要

MATLAB には一部の関数でマルチスレッドで処理されるものもありますが、並列処理のためのアドオン製品である Parallel Computing Toolbox™を使うことで MATLAB プログラムや Simulink® シミュレーションを並列で実行できます⁶。さらに MATLAB Parallel Server を使うことでクラスターマシンへスケールアウトすることもできます。複数の CPU コアや GPU を使った並列処理は様々な分野に適用できます(第2図)。

⁴ <https://jp.mathworks.com/support/requirements/language-interfaces.html>

⁵ <https://jp.mathworks.com/help/matlab/external-language-interfaces.html>

⁶ <https://jp.mathworks.com/solutions/parallel-computing.html>

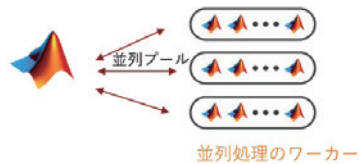


第2図: MATLAB のスケールアウトのイメージ図

クラスターサーバーでの MATLAB の並列処理は、対話形式の並列処理と、バッチでのオフロード形式の並列処理の2種類があります。対話形式の並列処理の例は下記の myparfor.m の通りで、parfor を実行した際に第3図のように MATLAB と、並列処理を行うクラスターサーバーの MATLAB プロセス(ワーカー)との間に並列プール(parpool)のセッションが張られ、タスクやデータが双方向に通信されて処理されます。

myparfor.m

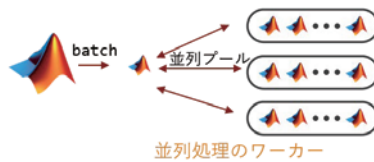
```
n = 200;
A = 500;
a = zeros(1,n);
parfor k = 1:n
    a(k) = max(abs(eig(rand(A)))));
end
```



第3図: 対話形式の並列処理

また、バッチでのオフロード形式の並列処理の例は下記のコードの通りで、batch コマンドで上記の myparfor.m を指定し、Pool オプションで並列数を指定することで、第4図のようにクラスターサーバーの内部だけで並列プールが張られて parfor の並列処理が実行されます。

```
job = batch('myparfor', 'Pool', 3);
wait(job)
out = fetchOutputs(job);
out = out{1};
```



第 4 図： バッチでのオフロード形式の並列処理

4. Oakbridge-CX と Wisteria/BDEC-01 での MATLAB の並列処理

東京大学情報基盤センターの Oakbridge-CX と Wisteria/BDEC-01 の Aquarius ノード群で MATLAB 及び MATLAB Parallel Server が使用できます。

クラスターサーバーで並列処理を行う場合、ジョブスケジューラとの連携が必須になりますが、MATLAB Parallel Server では付属する MATLAB ジョブスケジューラを使う方法以外に、サードパーティのジョブスケジューラと連携させることも可能です。Oakbridge-CX と Wisteria/BDEC-01 ではともに富士通 Technical Computing Suite (TCS) のジョブスケジューラと連携しています。MATLAB から 3. で紹介した並列処理を実行すると、自動的にジョブスケジューラのジョブ投入の скриプトが生成され、ジョブスケジューラでジョブが実行され、MATLAB Parallel Server の並列処理のためのプロセス(ワーカー)が処理を実行します。

```

>> deeplearningSample
labelCount =
    10x2 table
      Label    Count
    -----
         0     1000
         1     1000
         2     1000
         3     1000
         4     1000
         5     1000
         6     1000
         7     1000
         8     1000
         9     1000

ans =
    28    28

Starting parallel pool (parpool) using the 'GPU R2021b' profile ...
additionalSubmitArgs =
    '-L gpu=1 -q qt00 -L rsocsp=lecture-a -m b -m e -L proc-urpcc=16384'
Connected to the parallel pool (number of workers: 1).
accuracy =
    0.9876
Parallel pool using the 'GPU R2021b' profile is shutting down.

```

第 5 図： Wisteria/BDEC-01 で対話形式の並列処理を実行した結果

ログインノードで MATLAB をデスクトップ無しで起動し、ディープラーニングの学習を行うスクリプトを実行すると、自動的にジョブスケジューラのジョブ投入スクリプトが生成・実行され、計算ノード (Wisteria Aquarius) で GPU を使ってディープラーニングの学習が行われます。

Oakbridge-CX と Wisteria/BDEC-01 では、MATLAB の並列処理ができる他、2. に記述した他のプログラミング言語との連携を行った処理も可能です。MATLAB を使用する具体的な手順については、利用支援ポータルにあるドキュメント「MATLAB ジョブ連携手順書」を参照ください。

「MATLAB」提供開始：「計算・データ・学習」融合、 AI-for-HPC/AI-for-Science の実現へ向けて

中島 研吾

東京大学情報基盤センター

1. MATLAB 導入

いささか旧聞に属するが、当センターでは本年2月25日より MATLAB¹の提供を開始した。Oakbridge-CX² 及び Wisteria/BDEC-01 (Aquarius)³で利用することができる。利用は大学、短大、大学校、高専等の学位を授与する教育機関に所属するいわゆるアカデミックユーザーに限定されており、企業や研究機関に所属の方はご利用できないこと、予めご了承をお願いする次第である。客員研究員等の身分を有する場合は利用可能な場合もあるため、個別にご相談いただきたい (uketsuke(at)cc.u-tokyo.ac.jp 宛て)。

2. Wisteria/BDEC-01・h3-Open-BDEC による「計算・データ・学習」融合、 AI-for-HPC/AI-for-Science の実現

東京大学情報基盤センターで2021年5月に運用を開始した「Wisteria/BDEC-01⁴」は、「シミュレーション (Simulation)・データ (Data)・学習 (Learning) (S+D+L)」融合を目指す BDEC システム構想 (Big Data & Extreme Computing) に基づくシステムの第1号機であり、シミュレーションノード群 (Odyssey) とデータ・学習ノード群 (Aquarius) の2つの計算ノード群を有したシステムである (図1)。シミュレーションノード群 (Odyssey) とデータ・学習ノード群 (Aquarius) は、合計160本の InfiniBand EDR (100Gbps) を用いて2.0TB/秒のネットワークバンド幅で結合されている。また、データ・学習ノード群 (Aquarius) は合計800Gbpsのネットワーク転送速度で外部との通信が可能である。Aquariusの一部のノードは SINET 等の外部ネットワークを介して、サーバー、ストレージ、センサーネットワークを含む様々な外部リソースに直接アクセス可能であり、観測データをリアルタイムに取り込んで解析、シミュレーションに利用することも可能である。

当センターで開発した「ppOpen-HPC (自動チューニング機構を有するアプリケーション開発・実行環境)⁵」, 「h3-Open-BDEC (「計算+データ+学習」融合のための革新的ソフトウェア基盤)⁶」を利用し、高性能なアプリケーションを容易に開発することが可能である。計算科学シミュレーションは多くの場合、非線形な問題を扱うため、多数のパラメータスタディが必要となる。Wisteria/BDEC-01 では、機械学習による最適パラメータ推定を、外部から取り込んだ実験・観測データによる同化と組み合わせ、正確な解をより短時間で求めることを目指している (図2)。h3-Open-BDEC はそのようなアプリケーションの開発を支援する機能とともに、シミュレーションノード群 (Odyssey) とデータ・学習ノード群 (Aquarius) が協調したワーク

¹ <https://jp.mathworks.com/products/matlab.html>

² <https://www.cc.u-tokyo.ac.jp/supercomputer/obcx/service/matlab.php>

³ <https://www.cc.u-tokyo.ac.jp/supercomputer/wisteria/service/matlab.php>

⁴ <https://www.cc.u-tokyo.ac.jp/supercomputer/wisteria/service/>

⁵ <https://github.com/Post-Peta-Crest/ppOpenHPC>

⁶ <http://nkl.cc.u-tokyo.ac.jp/h3-Open-BDEC/>

ロードの実行を支援する機能も提供している [参考文献 1,2]。

これらのソフトウェア (h3-Open-SYS/WaitIO-Socket, h3-Open-UTIL/MP) は 2022 年 6 月から Wisteria/BDEC-01 で利用可能となっており, Odyssey と Aquarius を連携させた実行が可能となっている (図 3, 図 4)。詳細は [参考文献 3-6] をご覧いただきたい。

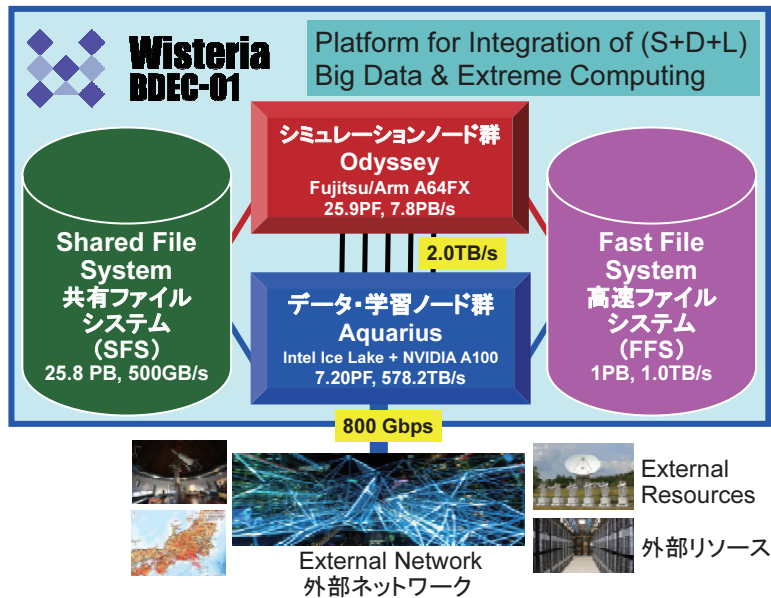


図 1 Wisteria/BDEC-01 の概要



図 2 Wisteria/BDEC-01 利用による「計算・データ・学習」融合のイメージ [1,2]

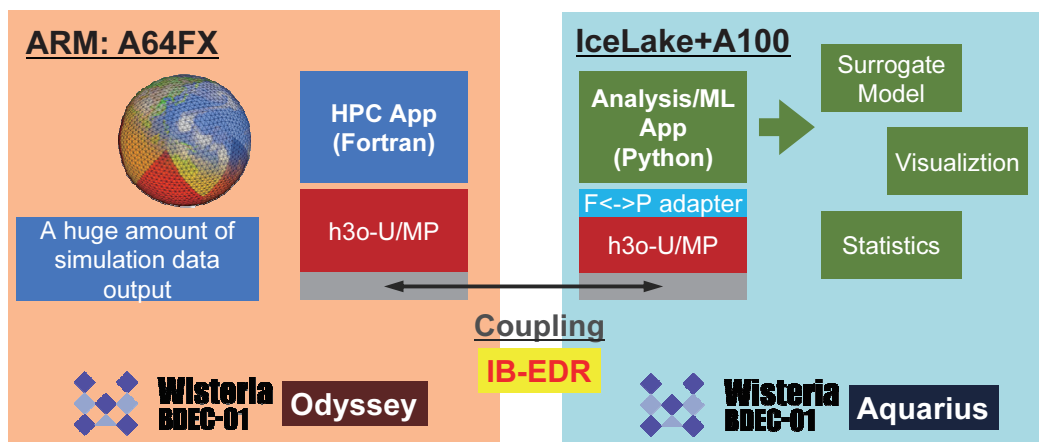


図3 h3-Open-UTIL/MP による，Odyssey（計算科学）・Aquarius（機械学習）連携ワークロードの例 [1,2]

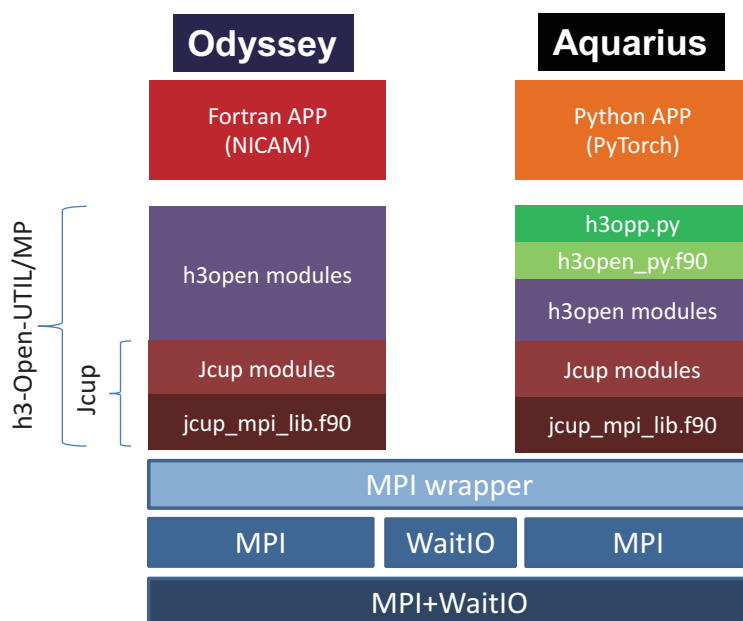


図4 h3-Open-UTIL/MP・h3-Open-SYS/WaitIO-Socket と MPI を併用した異機種間結合プログラムの構成 [1,2]

3. MATLAB への期待

MATLAB は数値計算，データ解析，機械学習などの様々な機能をサポートし，アルゴリズム開発やモデル作成に使用されている。Fortran, C/C++, Python 等で記述したプログラムから呼び出すことができる他，様々な CPU, GPU, それらのクラスタでも使用することが可能である。

当センターの利用者のほとんどは，自作，またはオープンソースのプログラムを実行している。今回の MATLAB 導入にあたっては，このようなユーザーの特性を考慮して，プログラムから MATLAB を呼び出す機能に特に着目した。

MATLAB のデータ解析，機械学習に関連する豊富な機能と，ユーザーの計算科学シミュレーションコードを組み合わせることによって，「計算・データ・学習」融合，

AI-for-HPC/AI-for-Science の実現に資するものと期待している。MATLAB は Odyssey では利用できないものの、前節で紹介した h3-Open-UTIL/MP, h3-Open-SYS/WaitIO-Socket 等の h3-Open-BDEC のソフトウェア群を使用すれば、Odyssey で稼働するプログラムから、MATLAB を呼び出す Aquarius 上のプログラムと連携させることも容易である。また、h3-Open-BDEC は様々な環境で動作するため、MATLAB と組み合わせるソフトウェア及び「計算・データ・学習」融合、AI-for-HPC/AI-for-Science の考え方の普及も実施していく予定である。

当センターでは、2020 年度より萌芽共同研究公募課題「AI for HPC : Society 5.0 実現へ向けた人工知能・データ科学による計算科学の高度化（試行）」⁷を実施している。本年度から随時募集となっているが、特に MATLAB を活用した「計算・データ・学習」融合、AI-for-HPC/AI-for-Science の事例を募集している。ご興味ある方（現ユーザーには限定せず）は、是非 [uketsuke\(at\)cc.u-tokyo.ac.jp](mailto:uketsuke(at)cc.u-tokyo.ac.jp) 宛てご一報いただきたい。

参 考 文 献

- [1] 住元真司, 荒川隆, 坂口吉生, 松葉浩也, 八代尚, 塙敏博, 中島研吾, WaitIO-Socket : 異種システム上の複数 MPI プログラムを結合する通信ライブラリの試作, 情報処理学会研究報告 (2021-HPC-181-7), 2021
- [2] 中島研吾, 古村孝志, 鶴岡弘, 松葉浩也, 坂口吉生, 住元真司, 笠井良浩, 池田輝彦, 八代尚, 荒川 隆, 塙敏博, 観測データ同化による長周期地震動リアルタイム予測へ向けた試み, 情報処理学会研究報告 (2021-HPC-182-08), 2021
- [3] 住元真司, 坂口吉生, 松葉浩也, 中島研吾, Wisteria/BDEC-01 利用事例 (3) データ受け渡しライブラリ h3-Open-SYS/WaitIO (1/2), スーパーコンピューティングニュース (東京大学情報基盤センター), Vol.24-2, 2022
https://www.cc.u-tokyo.ac.jp/public/VOL24/No2/10_202203Wisteria-1.pdf
- [4] 住元真司, 坂口吉生, 松葉浩也, 中島研吾, Wisteria/BDEC-01 利用事例 (4) データ受け渡しライブラリ h3-Open-SYS/WaitIO (2/2), スーパーコンピューティングニュース (東京大学情報基盤センター), Vol.24-3, 2022
https://www.cc.u-tokyo.ac.jp/public/VOL24/No3/12_202205-Wisteria-1.pdf
- [5] 荒川 隆, 八代 尚, 中島研吾, Wisteria/BDEC-01 利用事例 (5) マルチプログラム連成ライブラリ h3-Open-UTIL/MP (1/2), スーパーコンピューティングニュース (東京大学情報基盤センター), Vol.24-3, 2022
https://www.cc.u-tokyo.ac.jp/public/VOL24/No3/13_202205-Wisteria-2.pdf
- [6] 荒川 隆, 八代 尚, 住元真司, 中島研吾, Wisteria/BDEC-01 利用事例 (5) マルチプログラム連成ライブラリ h3-Open-UTIL/MP (2/2), スーパーコンピューティングニュース (東京大学情報基盤センター), Vol.24-4, 2022 (本号に掲載予定)

⁷ <https://www.cc.u-tokyo.ac.jp/guide/exploratory/AIforHPC/>

スーパーコンピュータシステム「大規模 HPC チャレンジ」課題募集のお知らせ

Wisteria/BDEC-01、Oakbridge-CX スーパーコンピュータシステムでは、「大規模 HPC チャレンジ」を実施しています。「大規模 HPC チャレンジ」は、スーパーコンピュータシステムがもつ最大規模のノード数を、最大 24 時間・1 研究グループで計算資源の専有利用ができる公募型プロジェクトです。採択条件等については、以下をご覧ください。皆様からの課題応募をお待ちしております。

※ 新型コロナウイルス感染症の状況次第で実施時間や実施条件の変更、中止の可能性もあります。

1. 提供資源

以下のスーパーコンピュータシステムのノードを最大 24 時間専有利用することができます。

- Wisteria/BDEC-01 スーパーコンピュータシステムのシミュレーションノード群 (Odyssey) 6,144 ノード (294,912 コア)、データ・学習ノード群 (Aquarius) 36 ノード (GPU 288 基)
- Oakbridge-CX スーパーコンピュータシステムの計算ノード 1,280 ノード (内 SSD 搭載 112 ノード)

2. 利用案内

- 1 ヶ月に 1 回、原則として月末処理前日の 9:00～翌 9:00 までの最大 24 時間、提供資源を専有利用することが可能です。

(大規模 HPC チャレンジの当日に、チャレンジ用の設定に変更いたします。この変更は 1 時間程度の作業時間が見込まれるため、大規模 HPC チャレンジのご利用時間は 10 時頃開始、翌日 9 時終了となります。)

- 課題は公募制とし、現ユーザーに限定せず、広く課題を募集します。個人、及びグループによる応募が可能です。各月に 1 グループの採用 (*) を原則とします。
- 本制度により得られた成果については公開して頂きます。**成果公開には東京大学情報基盤センターのスーパーコンピュータシステムを利用し、「大規模 HPC チャレンジ」制度によって実施した旨を明記していただきます。また、「スーパーコンピューティングニュース」や広報誌等への成果報告記事の執筆などを行っていただきます。
- センターの主催、共催するセミナー、ワークショップ等でご発表いただく場合があります。
- 利用料金は無料です。

*Wisteria/BDEC-01 においては各月 Odyssey で 1 件、Aquarius で 1 件、最大 2 件まで受入可能、ただし 1 グループで Odyssey、Aquarius 両方利用することも可能

3. 実施日程

2022 年度の今後の「大規模 HPC チャレンジ」実施日程は表 1～2 のとおりです。

※ 新型コロナウイルス感染症の状況次第で、実施時間や実施条件の変更、中止となる可能性もあります。

表 1. 2022 年度 Wisteria/BDEC-01 大規模 HPC チャレンジ実施日程

実施日時		募集締切	審査	採択通知
第 3 回	2022 年 12 月 22 日(木) 10:00 ～ 23 日(金) 9:00	2022 年 10 月 31 日 (月) 17:00 【締切】	11 月上旬	11 月中旬
	2023 年 1 月 26 日(木) 10:00 ～ 27 日(金) 9:00			
	2023 年 2 月 21 日(火) 10:00 ～ 22 日(水) 9:00			
	2023 年 3 月 30 日(木) 10:00 ～ 31 日(金) 9:00			

表 2. 2022 年度 Oakbridge-CX 大規模 HPC チャレンジ実施日程

実施日時		募集締切	審査	採択通知
第 3 回	2022 年 12 月 20 日(火) 10:00 ~ 21 日(水) 9:00	2022 年 10 月 31 日 (月) 17:00 【締切】	11 月上旬	11 月中旬
	2023 年 1 月 24 日(火) 10:00 ~ 25 日(水) 9:00			
	2023 年 2 月 16 日(木) 10:00 ~ 17 日(金) 9:00			
	2023 年 3 月 30 日(木) 10:00 ~ 31 日(金) 9:00			

- ・ メンテナンス等の都合により募集スケジュールが変更となることがあります。最新情報は Web Page¹をご覧ください。
- ・ 年複数回を申し込むことも可能ですが、申込状況によりご希望に添えない場合もありますのであらかじめご了承ください。また、一回の申し込みで利用可能なのは一回のみです。
- ・ 表に掲載されている以外の日程でも募集を行うことがあります。最新情報は Web Page¹をご覧ください。

4. 研究対象

「大規模 HPC チャレンジ」では、提供する最大ノードを使用する大規模計算を実施する研究に限定します。申込者及び研究グループのメンバーは、国内外の並列計算機を利用した大規模計算の実績があることを前提とし、以下のような「High-Performance Computing」に関連した幅広い分野の研究を対象としています。

- ・ 大規模シミュレーション
- ・ 大規模データ処理
- ・ 大規模ベンチマーク、演算・通信システム性能評価
- ・ その他、大規模計算に関係するソフトウェア実行

5. 利用資格

利用資格は、申込書を基に、東京大学情報基盤センタースーパーコンピューティング研究部門の教職員、及び、外部委員により構成される審査委員会において審査されます。現ユーザーである必要はありません。

応募課題は、審査委員会により採択課題を選考し、できるだけ速やかに公表を行う予定です。

なお、申込者は「国内の大学、公共機関に所属する研究者、及び民間企業に所属する者」とします。また、研究グループのメンバー又は申込者が企業の方の場合は、以下の書類のいずれかを提出していただく必要があります。

- ・ 「共同研究契約書の写し」
(申込者の所属機関と共同研究契約を結んでいる研究組織に所属する者)
- ・ 「適切に監督を行うことを記した誓約書及び請負契約書の写し」
(申込者の所属機関と請負契約を結んでいる企業の従業員)
- ・ 「利用規定にある利用目的を遵守することを記した誓約書」
(民間企業に所属している者)

6. 採択基準、審査方法

応募課題は、以下の基準により、東京大学情報基盤センタースーパーコンピューティング研究部門の教職員、及び、外部委員より構成される審査委員会により採択課題を選考し、できるだけ速やかに公表を行う予定です。

主な採択基準

- 計算・結果の詳細を論文等も含めて公表できること。
- 計算結果が科学的に有用、あるいは社会的なインパクトがあると考えられること。

¹ 「大規模 HPC チャレンジ」

<https://www.cc.u-tokyo.ac.jp/guide/hpc/>

- C) 提供する最大ノード数の利用を目標としていること。
- D) 計画に実現性があり、短期間で効果を示すことが可能であること（一回の使用時間は最大 24 時間です）。
- E) 本システムの運用、ユーザーにとって有用な情報を提供すること。

※ 項目 A) ～ D)は、必須となります。項目 E)は必須ではありませんが、申込書に該当する記述がある場合、加
点評価される場合があります。

7. 利用申込

募集要項、スーパーコンピューターシステム利用規程等をよくお読みの上、申込書及び研究グループメンバー情報
に必要事項をご記入ください。ご記入頂いた申込書及び研究グループメンバー情報、論文は以下の提出先まで、電子
メールでお送りください。

申込書に必要な項目・提出書類は以下の通りです。

1. 申込年月日
2. 利用希望時期
3. 申込者情報（氏名、所属、職名、連絡先住所、E-mail、電話）
4. 研究課題名（和文、英文）、概要
5. 研究課題の内容、目標
6. 申込者、研究グループメンバーの当該分野における研究業績のうち、大規模計算機利用の実績として代表的な
論文 1 編の別刷り
7. プログラム情報、利用スケジュール等
8. 要望事項、特記事項
9. 研究グループメンバーの情報

8. 問い合わせ先

申込等ご不明な点は、電子メールでお問い合わせください。

（電話でのお問い合わせはご遠慮ください。なお、詳細は本センターWeb Page¹でもご案内しておりますので、
あわせてご覧ください。）

【 提出先 】

E-Mail : koubo@cc.u-tokyo.ac.jp
東京大学 情報システム部 情報戦略課 研究支援チーム

【 問い合わせ先 】

E-Mail : uketsuke@cc.u-tokyo.ac.jp
東京大学 情報システム部 情報戦略課 研究支援チーム

研究成果の登録のお願い

情報戦略課研究支援チーム
情報基盤センタースーパーコンピューティング部門

研究成果の登録は、本センターのスーパーコンピュータシステムを利用して得られた研究成果のうち、論文、口頭発表、著書、受賞情報についてご報告いただくものです。研究成果の登録は、本センタースーパーコンピューティング部門の Web サイト (<https://www.cc.u-tokyo.ac.jp/>) から「研究成果登録」に進んでください。なお、ご報告いただいた内容は、研究成果データベースへの登録、本センター発行の広報誌及び Web ページに掲載させていただきますので、ご了承ください。

研究成果は、東京大学におけるスーパーコンピュータシステムの整備・拡充につながるものとなりますので、利用者の皆様には何卒ご協力くださいますようお願い申し上げます。

The screenshot shows the website of the Supercomputing Division, Information Technology Center, The University of Tokyo. The navigation bar includes links for Home, System, Utilization, Support, FAQ, Research Events, Publications, Research Results, Research Department, Research Activities, and Wisteria/BDEC-01. The 'Research Results' link is circled in green and labeled ①. Below the navigation bar, the 'Research Results' section is highlighted, and the 'Research Results Registration' link is circled in green and labeled ②. A large blue arrow points down to the 'Research Results Registration' page. On this page, the 'Research Results Registration' link is circled in green and labeled ③. Below this, the 'Research Results Registration' form is shown, with the 'Login' button circled in green and labeled ④. The form includes fields for Account Name, Email Address, and a dropdown for the type of research result (e.g., Paper, Oral Presentation, Book, Award). The 'Login' button is labeled 'ログイン'.

① 「研究成果」
→「研究成果の取扱い」
をクリック

② 「研究成果の取扱い」ページ下部の「研究成果の登録について」→「研究成果登録のページ」をクリック

③ 「アカウント名(利用者番号)」及び「メールアドレス」を入力し、登録したい業績を選択

④ 「ログイン」をクリックし、成果登録ページで研究成果の登録をお願いいたします

4・5月のジョブ統計

1. Wisteria/BDEC-01スーパーコンピュータシステム(Odyssey) ジョブ処理状況 (RedHat Enterprise Linux 8)

年月	登録者数	実利用者数	処理件数				接続時間 [時間]	ファイル使用量 [GiB]		ログイン (実CPU)	演算時間 [ノード時間] (経過時間)			平均ノード 利用数 (ノード)	ノード 利用率 (%)
			ログイン	プリポスト	インタラクティブ ジョブ	バッチジョブ		/home	/lustre		プリポスト	インタラクティブ ジョブ	バッチジョブ		
2022年4月	1,687	456	6,822	184	241	13,256	30.076	1,225	1,203,879	8,038	366	45	703,985	1,139.2	14.8
5月	1,794	506	8,152	228	647	38,581	55.029	1,312	1,387,835	2,068	578	73	1,114,477	1,524.7	19.9
2021年5月	860	206	3,348	54	188	6,718	13.623	148	66,121	697.32	40	35	336,219	822.0	10.7
6月	974	270	5,845	97	186	12,367	23.491	361	189,018	1,398.60	122	36	606,566	842.5	11.0
7月	1,103	229	4,638	66	64	10,730	24.723	438	211,696	1,670.10	74	10	996,901	1,492.2	19.4
8月	939	147	2,212	82	28	4,551	9.866	383	128,337	2,218.48	255	6	123,322	219.4	2.9
9月	1,058	196	3,835	76	150	20,431	19.769	434	246,063	1,944.79	281	40	204,279	289.0	3.8
10月	1,193	311	5,221	256	226	12,759	27.589	580	396,240	836.51	284	85	351,435	480.9	6.3
11月	1,279	256	5,337	76	311	7,340	26.676	774	515,631	3,092.91	86	91	147,549	208.8	2.7
12月	1,286	298	6,189	76	366	9,752	31.952	837	833,732	1,879.83	123	64	672,591	920.1	12.0
2022年1月	1,323	294	7,329	92	220	27,108	33.550	961	979,806	1,926.47	256	60	801,198	1,096.0	14.3
2月	1,293	269	6,015	168	275	13,449	29.553	1,105	1,021,929	2,675.51	496	124	813,968	1,235.3	16.1
3月	1,221	265	6,896	135	187	60,265	31.203	1,131	1,007,327	9,642.52	322	42	1,222,254	1,715.7	22.3
合計			68,491	1,536	2,901	230,589	343.477			37,392	3,243	676	7,758,525		

・試運転は、2021年5月14日より開始。正式サービスは、2021年8月2日より開始

・接続時間：ログイン時間の累計

・ログイン(実CPU)：コア時間単位

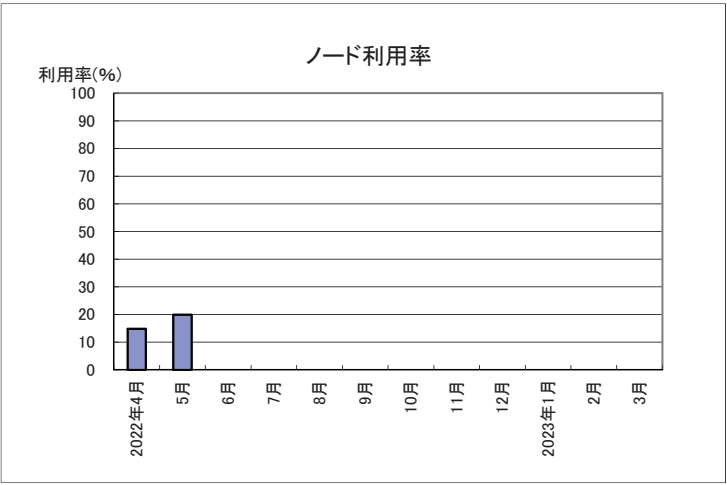
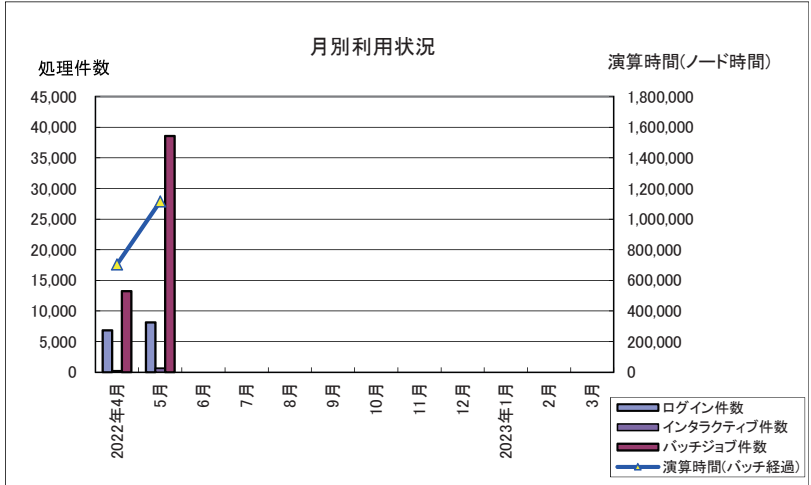
・2021年5月分は合計に含まない

・ノード利用率：インタラクティブおよびバッチジョブの経過時間を1ノードが100%動作したと仮定した場合の利用ノード数。

計算式＝1ヶ月のインタラクティブおよびバッチジョブ経過時間合計÷1ヶ月の稼働時間

・ノード利用率：サービスノードに対する利用率。計算式＝ノード利用率÷サービスノード数×100

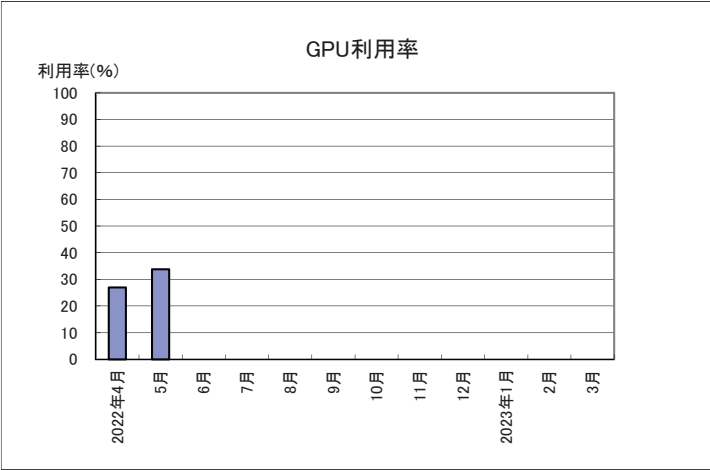
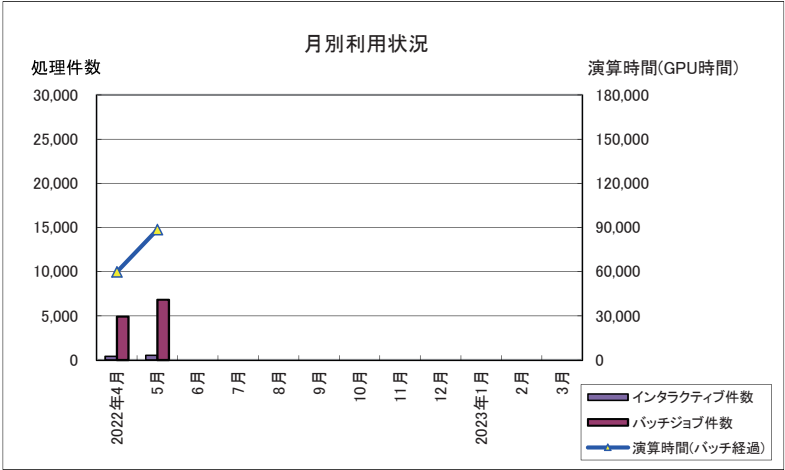
・下線部分の値に誤りがございました。お詫びして訂正いたします。



2. Wisteria/BDEC-01スーパーコンピュータシステム(Aquarius) ジョブ処理状況 (RedHat Enterprise Linux 8)

年月	処理件数		演算時間 [GPU時間](経過時間)		平均GPU 利用数 (GPU)	GPU 利用率 (%)
	インタラクティブ ジョブ	バッチジョブ	インタラクティブ ジョブ	バッチジョブ		
2022年4月	422	4,911	277	59,779	97.2	27.0
5月	541	6,831	338	88,557	121.6	33.8
2021年5月	252	5,095	158	35,646	87.5	24.3
6月	953	21,394	1,245	101,956	143.3	39.8
7月	1,540	15,155	1,531	117,749	180.9	50.3
8月	466	3,004	470	37,677	67.9	18.9
9月	480	3,599	358	45,671	65.1	18.1
10月	272	3,139	91	63,871	87.5	24.3
11月	1,122	5,824	1,107	63,864	91.9	25.5
12月	1,077	7,209	957	107,582	148.5	41.2
2022年1月	615	8,846	623	100,024	137.7	38.2
2月	863	6,116	801	96,072	147.0	40.8
3月	479	7,204	417	108,978	150.1	41.7
合計	8,830	93,232	8,215	991,780		

- ・試運転は、2021年5月14日より開始。正式サービスは、2021年8月2日より開始
 - ・登録者数、実利用者数、ログイン件数、接続時間、ファイル使用量、ログイン(実CPU)はWisteria/BDEC-01(Odyssey) と共通。
 - ・2021年5月分は合計に含まない
- ・GPU利用数： インタラクティブおよびバッチジョブの経過時間を1GPUが100%動作したと仮定した場合の利用GPU数。
計算式＝1ヶ月のインタラクティブおよびバッチジョブ経過時間合計÷1ヶ月の稼動時間
 - ・GPU利用率： サービスGPUに対する利用比率。 計算式＝GPU利用数÷サービスGPU数×100
 - ・下線部分の値に誤りがございました。お詫びして訂正いたします。



3. Oakbridge-CX スーパーコンピュータシステムジョブ処理状況 (Red Hat Enterprise Linux 7、CentOS 7)

年月	登録者数	実利用者数	処理件数				接続時間 [時間]	ファイル使用量 [GiB]		ログイン (実CPU)	演算時間 [ノード時間] (経過時間)			平均ノード 利用数 (ノード)	ノード 利用率 (%)
			ログイン	プリポスト	インタラクティブ ジョブ	バッチジョブ		/home	/work		プリポスト	インタラクティブ ジョブ	バッチジョブ		
2022年4月	1,095	312	6,123	10	310	75,245	24,901	1,461	1,384,480	3,981.48	0	112	389,831	630.5	46.1
5月	1,185	324	12,165	0	851	248,400	40,747	1,465	1,423,688	2,470.79	0	254	597,464	818.0	59.8
2021年5月	1,196	299	6,395	0	982	60,221	38,699	874	1,292,853	8,996.00	0	992	513,767	704.0	51.5
6月	1,087	358	8,649	43	1,356	79,376	48,740	993	1,357,480	2,853.51	1	1,093	587,184	832.8	60.9
7月	1,132	366	8,545	42	1,860	62,710	57,857	1,017	1,430,169	16,807.20	2	674	666,144	909.4	66.5
8月	1,141	278	5,494	59	1,814	31,559	22,299	1,060	1,499,355	4,136.73	5	592	524,972	945.0	69.1
9月	1,067	284	6,700	17	1,585	31,851	32,068	1,095	1,592,894	2,314.47	0	592	646,063	1,015.7	74.2
10月	1,120	305	8,969	29	3,898	72,268	48,565	1,183	1,697,798	3,616.60	3	1,253	772,159	1,057.7	77.3
11月	1,179	348	8,683	0	7,370	63,788	49,289	1,227	1,860,386	3,446.55	0	881	740,267	1,048.5	76.6
12月	1,187	362	9,052	2	763	73,500	43,544	1,377	1,883,279	2,295.56	0	258	742,688	1,016.8	74.3
2022年1月	1,200	375	10,915	0	497	84,721	50,652	1,522	1,941,483	7,899.93	0	262	726,123	994.1	72.7
2月	1,229	328	7,404	0	521	70,221	38,572	1,507	1,842,598	2,126.96	0	192	609,550	925.1	67.6
3月	1,245	317	6,767	0	403	48,195	39,693	1,579	1,674,826	3,006.83	0	228	589,505	830.9	60.7
合計			99,466	202	21,228	941,834	496,927			54,957	11	6,391	7,591,950		

・接続時間：ログイン時間の累計

・ログイン(実CPU)：コア時間単位

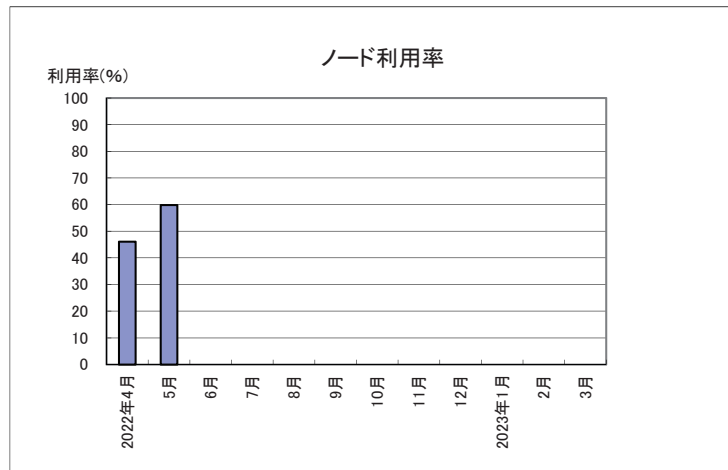
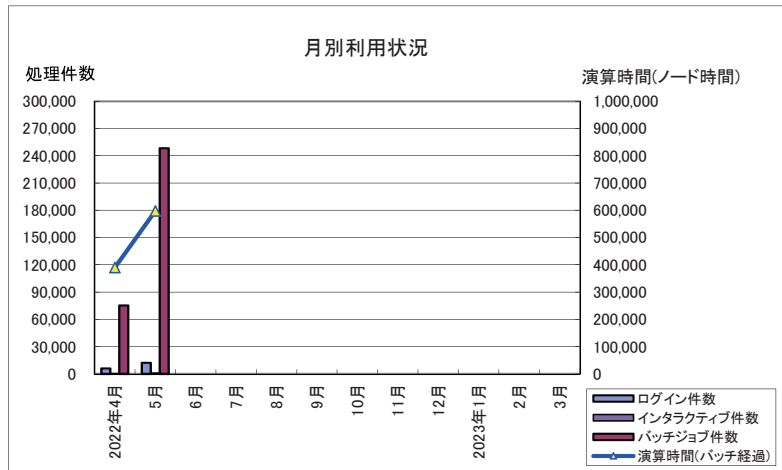
・2021年5月分は合計に含まない

・ノード利用数：インタラクティブおよびバッチジョブの経過時間を1ノードが100%動作したと仮定した場合の利用ノード数。

計算式＝1ヶ月のインタラクティブおよびバッチジョブ経過時間合計÷1ヶ月の稼働時間

・ノード利用率：サービスノードに対する利用率。計算式＝ノード利用数÷サービスノード数×100

・下線部分の値に誤りがございました。お詫びして訂正いたします。



Wisteria/BDEC-01 利用事例 (6)

マルチプログラム連成ライブラリ h3-Open-UTIL/MP (2/2)

荒川隆 (ClimTech)

八代尚 (国立環境研究所)

住元真司 (東京大学情報基盤センター)

中島研吾 (東京大学情報基盤センター)

1 はじめに

前回 (2022 年 5 月号) [1]と今回の 2 回に分けて複数のシミュレーションプログラムを連成実行し「計算・データ・学習」融合を推進するためのライブラリ h3-Open-UTIL/MP について紹介する。前回是一般的な単一アーキテクチャ環境で複数プログラムを連成するケースについて説明したが、今回は h3-Open-UTIL/MP の特徴的な機能である異機種間連成, Python APP 連成, アンサンブル連成機能などについて紹介する。連成計算の基本的な概念や実行方法については前号で詳述しているので、これらについて未経験の読者は先に前号の記事を参照されたい。また、異機種間連成を実現するために h3-Open-UTIL/MP はライブラリ h3-Open-SYS/WaitIO を利用しているが、h3-Open-SYS/WaitIO についてはスーパーコンピューティングニュース前々号と前号に紹介記事が掲載されている[2][3]ので併せて参照いただけると幸いである。h3-Open-SYS/WaitIO は記事掲載後も開発が続けられ、インターフェースなどに若干の仕様変更がなされている。そこで、本稿では h3-Open-SYS/WaitIO の最新情報についても記述する。なお前回と同様、このような計算方法に対しては基本的に「連成計算」の語を用い、一部はコミュニティの慣習に従って結合 (例: 大気海洋結合) を用いることとする。また以下では h3-Open-SYS/WaitIO は単に WaitIO と表記する。

2 異機種間連成機能

2.1 異機種間連成の概要

東京大学情報基盤センターの Wisteria/BDEC-01 をはじめ、名古屋大学の不老や海洋研究開発機構の地球シミュレータなど、近年はアーキテクチャの異なる複数のサブシステムで構成される計算機が増えている。これは現代のスーパーコンピューティングの対象領域がシミュレーションにとどまらず、機械学習や大規模データ処理など要求性能の異なる多様なタスクに拡大していることに対応するためである。これらのサブシステム同士は基本的にはファイル共有によって計算+データ+学習連携機能を実現しているが、サブシステム間でのリアルタイムな連成計算が可能になれば、シミュレーションと解析の同時実行やシミュレーションへの機械学習の反映など、ファイルを経由した逐次的な処理では対応できなかった計算手法が実現できる。このような背景から WaitIO が開発され、別掲記事で紹介されているように観測データ同化による長周期地震動リアルタイム予測などに利用されている。この異機種間通信機能を h3-Open-UTIL/MP に組み込む事により時空間構造の異なる複数のプログラムを異機種間で連成することが可能となる。

2.2 WaitIO の最新情報

前節で述べたように WaitIO は現在も精力的に開発が進んでおり[3]の記事掲載後も使用方法に若干の変更がある。そこで本節では WaitIO の最新情報について説明する。前号記事掲載後に行われたアップデートのうち利用者に直接関係するのは WaitIO が Environmental Modules に対応したことである。Wisteria/BDEC-01 では環境設定に Environment Modules を用いており、コンパイラの切り替えなどを module コマンドで行うようになっている。WaitIO の設定を Environment Modules で行う場合の具体的なコマンドはごく一般的な module load waitio であるが、注意すべき点として、異機種間通信ライブラリという WaitIO の役割に対応するため使用されるコンパイラや MPI に応じてコマンドで設定される内容が変化するということがある。Wisteria/BDEC-01 の Odyssey で WaitIO を用いる場合のコマンドは図 1 のようになる¹。このときに設定される環境変数は図 2 に示すとおりである。一方、Aquarius で WaitIO を用いる場合のコマンドと設定される環境変数は図 3、図 4 に示すようになる。WaitIO を利用する際は複数の環境でプログラムをコンパイル・リンクする必要があるため、コンパイル・リンク環境に応じて適切なコマンドを用いなければならない。また、リンクすべきライブラリも環境に応じて複数あるため適切なライブラリを選択する必要がある。具体的には Odyssey では A64FX 用のライブラリである libwaitio_a64fx.so を、Aquarius では用いるコンパイラや MPI が Intel+impi か gnu+ompi かに応じて libwaitio_intel.so もしくは libwaitio.so をリンクするようにする。

```
$module purge
$module load odyssey
$module load waitio
```

図 1 Odyssey での WaitIO の module load 手順

```
WAITIO_INC=/work/opt/local/aarch64/apps/fj/1.2.35/fjmpi/1.2.35/waitio/1.0/include
WAITIO_SERVER_HOST=wisteria01
WAITIO_DIR=/work/opt/local/aarch64/apps/fj/1.2.35/fjmpi/1.2.35/waitio/1.0
WAITIO_LIB=/work/opt/local/aarch64/apps/fj/1.2.35/fjmpi/1.2.35/waitio/1.0/lib
```

図 2 Odyssey で設定される WaitIO の環境変数

```
$module purge
$module load intel
$module load impi
$module load waitio
```

図 3 Aquarius での WaitIO の module load 手順

```
WAITIO_INC=/work/opt/local/x86_64/apps/intel/2021.2.0/impi/2021.2.0/waitio/1.0/include
WAITIO_SERVER_HOST=wisteria01
WAITIO_DIR=/work/opt/local/x86_64/apps/intel/2021.2.0/impi/2021.2.0/waitio/1.0
WAITIO_LIB=/work/opt/local/x86_64/apps/intel/2021.2.0/impi/2021.2.0/waitio/1.0/lib
```

図 4 Aquarius で設定される WaitIO の環境変数

¹ なお、コンパイラや MPI 環境が load されていない状態で module load waitio コマンドを実行しても無効となる。

実行に際しても図5のように `module load` コマンドで適切な WaitIO 環境を設定してやる必要がある。なお図5右図の例では、実行バイナリを `gcc+impi` で生成しているため、`intel+impi` で WaitIO 環境を設定した後、改めて `gcc+impi` を load している。

<pre>#!/bin/bash #PJM -L rscgrp=regular-o 省略 module load fj module load fjmp module load waitio export WAITIO_MASTER_HOST=`hostname` export WAITIO_MASTER_PORT=7100 export WAITIO_PPID=0 export WAITIO_NPB=2 waitio-serv-a64fx -d -m \$WAITIO_MASTER_HOST mpiexec -np 160 ./nicam</pre>	<pre>#!/bin/bash #PJM -L rscgrp=regular-a 省略 module unload aquarius module unload gcc module load intel module load impi module load waitio export WAITIO_MASTER_HOST=`waitio-serv -c` export WAITIO_MASTER_PORT=7100 export WAITIO_PPID=1 export WAITIO_NPB=2 module unload intel module unload impi module load gcc impi mpiexec -n 20 ./ada</pre>
---	--

図5 WaitIO を用いた異機種間連成の実行ジョブスクリプト（左図：Odyssey, 右図：Aquarius）

2.3 異機種間連成の実現方法

本節では h3-Open-UTIL/MP がどのようにして WaitIO と連携し異機種間連成を実現しているかを説明する。WaitIO は異機種間通信用の基本関数として `waitio_isend`, `waitio_irecv`, `waitio_wait` の3種類を提供している。これらは MPI の同等の関数と同じインターフェースを持ち、容易に相互変換可能である。一方、h3-Open-UTIL/MP はモデル間のデータ交換を `mpi_isend`, `mpi_irecv`, `mpi_wait` の3種の関数のみで実行しているため、これらを WaitIO の上記関数に置き換えるだけで異機種間データ交換に対応できる。一方、データ交換前の初期化プロセスではアプリケーション間の大域通信を行っているため、これを WaitIO で置き換える必要がある。そこで大域通信についてはアプリケーション内の通信を MPI で、アプリケーション間 (= 異機種間) の通信を WaitIO で行うようにした。全体のプログラム構造を図6に示す。図で右側の柱は Aquarius 上で Python のアプリケーションを実行する際のプログラム構造である。これについては次章で詳述する。異機種間連成に関わるプログラム構造は図の下部、青系の四角で示した3つの層である。最下層の MPI+WaitIO は大域通信を行うレイヤで、ここでは MPI と WaitIO を組み合わせて通信を実現している。具体的な方法は次に説明する。下から2段目の層は局所通信を行うレイヤである。ここでは、送受信の相手が別アプリケーションの場合は WaitIO を、同一アプリケーションの場合は MPI を用いて通信を行う。下から3段目の層は MPI と WaitIO を統合し単一の API を提供する Wrapper レイヤである。これによって、h3-Open-UTIL/MP 本体のコードを修正することなく、異機種間連成に対応させることが可能となり、同機種間連成専用プログラムとも容易に切り替え可能となる。

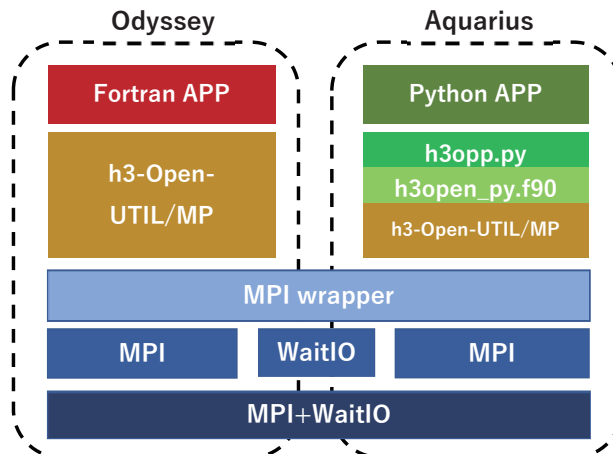


図 6 h3-Open-UTIL/MP+WaitIO のプログラム構造

次に上図最下段の MPI+WaitIO がどのようにして実現されているかを説明する。h3-Open-UTIL/MP が用いている大域通信用 MPI ルーチンは MPI_Bcast, MPI_Gather, MPI_Reduce, MPI_Allreduce の 4 種類に限られており、異機種間連成ではこれらのルーチンを WaitIO の基本通信関数と MPI の大域通信関数を組み合わせて実現すればよいことになる。この際、異機種間の大域通信を waitio_isend/irecv を用いた多対多通信で実装すると並列規模が大きくなった際に対応できなくなる懸念があったため、異機種間の通信は各アプリケーションの代表プロセス間のみの 1 対 1 通信で行うようにした。実装された通信方法を MPI_Gather に相当する大域通信を例として図 7 に示す。図で横に並んだ 3 つの四角がひとつのアプリケーションを、個々の四角が個々のプロセスを表している。左端の四角は各アプリケーションの代表プロセス(ゼロ番プロセス=King)である。図の黄色い四角が Gather 先のプロセス(root)とすると、はじめに MPI 関数で King に情報を集める。次いでアプリケーションの King 間を WaitIO で通信し、最後に MPI 関数で対象プロセスにデータを送る。図の例では Gather 先のプロセスを代表プロセス以外のプロセスとしたが、現実的には Gather 先プロセスは代表プロセスとなることがほとんどであるため最後の手続きは不要になる。

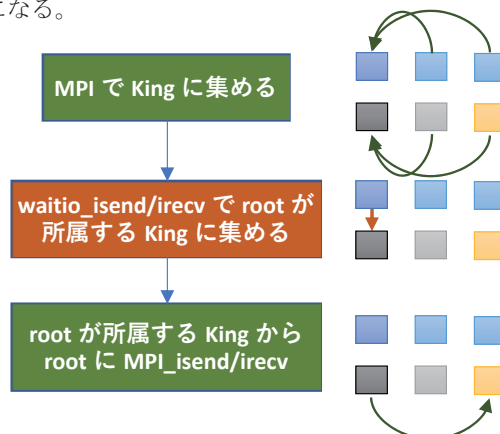


図 7 MPI+WaitIO を用いた大域通信の例 (Gather)

本節ではサンプルプログラムを参照しつつ、異機種間連成の具体的な手順について説明する。なお単一機種内連成については前号の記事[1]で詳述しているので、ここでは異機種間連成に関連する部分のみを取り上げる。h3-Open-UTIL/MP のライブラリと API は Wisteria/BDEC-01 上で公開されているが、単一機種用と異機種用ではライブラリが異なるので注意が必要である。また異機種用は Odyssey 用と Aquarius 用でも異なるライブラリを用いる必要がある。ライブラリの名称はいずれも同じ (libjcup4.a と libh3ou.a) でディレクトリによって使い分けようになっている。異機種間連成用のライブラリとインクルードファイルのディレクトリ構成は図 8 に示すとおりで、include と lib ディレクトリの下に odyssey と aquarius ディレクトリがあり、各々のシステム用のインクルードファイルとライブラリが格納されている。従って、Odyssey 用のバイナリを生成する際は odyssey ディレクトリを、Aquarius 用のバイナリを生成する際は aquarius ディレクトリをそれぞれ参照する。加えて、2.1 節で述べたように適切な WaitIO モジュールを load する。



```
>>>>>>>>>> h3ou_coupling_end IN  
<<<<<<<<<<< h3ou_coupling_end OUT
```

スーパーコンピューティングニュース

```
#How to make nicam and ada
cd src
export SYSTEM=odyssey
module purge
module load odyssey
module load waitio
make clean
make nicam

export SYSTEM=aquarius
module purge
module load intel impi
module load waitio
module unload intel impi
module load gcc omp
make clean
make ada
```

図9 異機種間連成サンプルプログラムのコンパイル手順

3 Python APP 連成

本章では h3-Open-UTIL/MP の拡張機能の一つである Python アプリケーションの連成機能について説明する。はじめに Python API の構造と仕様について説明し、次いで Python アプリケーションを連成するために必要な手順を具体例に基づきつつ説明する。

3.1 Python API の構造と仕様

h3-Open-UTIL/MP のコードは Fortran で記述されており、ネイティブな API は Fortran のモジュール形式で提供されている。Python は C 言語の関数を呼び出す機能を持ち C 言語との連携は容易であるが Fortran のモジュールを直接使用することはできない。しかしながら Fortran2003 以降では C 言語の互換インターフェースを提供する仕組みが標準装備されているため、この機能を使うことで Python から C 言語の関数として Fortran のサブルーチン（や関数）をコールすることができる。具体的な方法を h3-Open-UTIL/MP の初期化サブルーチンを例として説明する。h3-Open-UTIL/MP の初期化サブルーチン h3ou_init はモデル名と設定ファイル名（および optional 引数としてアンサンプル数）を与える仕様になっている。このサブルーチンを Python から呼び出せるように、モジュールに含まれない Fortran サブルーチンを定義する。初期化サブルーチンの例を図 10 に示す。C 言語は文字列の扱いが Fortran と異なっているため文字列を渡す際は先頭位置と長さの 2 つの引数を渡すことになる。この 2 つの値から関数 get_char_str で Fortran の文字列に変換し、h3-Open-UTIL/MP のネイティブ API である h3ou_init をコールする。このサブルーチンには bind(C) 属性が付加され、これによって Python からは C 言語の関数として呼び出すことが可能となる。初期化を含む各種 API ルーチンは図 6 の h3open_py.f90 に格納されている。次に呼び出し側である Python の関数を図 11 に示す。はじめに argtypes でインターフェースを定義し、ついで restypes で関数の返値を定義する。ここでは返値なし (c_void_p) が与えられている。渡されてきた引数は C 言語のポインタ型に変換され図 10 に示された Fortran サブルーチンに渡されるとともに、文字列の長さもポインタ引数として渡される。

```

subroutine h3oup_init(comp_name, comp_name_len, config_name, &
                    config_name_len) bind(C)

  use h3ou_py_base, only : get_char_str
  use h3ou_api, only : h3ou_init
  implicit none
  character(len=1)      :: comp_name(*)
  integer, intent(IN) :: comp_name_len
  character(len=1)      :: config_name(*)
  integer, intent(IN) :: config_name_len

  call h3ou_init(trim(get_char_str(comp_name, comp_name_len)), &
                trim(get_char_str(config_name, config_name_len)))

end subroutine h3oup_init

```

図 10 PythonAPP 連成用 Fortran API の内容例

```

def h3ou_init(comp_name, config_file_name):

    h3oupf.h3oup_init.argtypes = [
        ctypes.POINTER(ctypes.c_char),
        ctypes.POINTER(ctypes.c_int32),
        ctypes.POINTER(ctypes.c_char),
        ctypes.POINTER(ctypes.c_int32)
    ]
    h3oupf.h3oup_init.restype = ctypes.c_void_p

    comp      = ctypes.create_string_buffer(comp_name.encode("utf-8"))
    comp_len  = len(comp_name)
    config     = ctypes.create_string_buffer(config_file_name.encode("utf-8"))
    config_len = len(config_file_name)

    h3oupf.h3oup_init(comp, ctypes.byref(ctypes.c_int32(comp_len)), ¥
                      config, ctypes.byref(ctypes.c_int32(config_len)))

```

図 11 APython APP 連成用 API の内容例

3.2 Python APP 実行のための環境構築

Python の利点は、豊富なライブラリ群が全世界の開発者から提供されており、これらのライブラリを有効利用することで高度な機能を簡単に実現することができる、という点にある。Python アプリケーションを連成する際も目的に応じて適切なライブラリをインストールして利用することになるが、その際、システム環境にはインストールできずユーザー領域に環境を構築することになる。そのための手順は Wisteria/BDEC-01 システム利用手引き書、最新版の手引き書 (Version 1.1.9) では 192 ページ「9.2.5.追加パッケージのインストール方法」に記載されている。この手順に従って /work/groupname/username/package_name にインストールされたライブラリ群はジョブスクリプト内で、

```
source /work/groupname/username/package_name/bin/activate
```

と記述することでジョブに投入された Python プログラムから参照可能となる。

該分野では一般的な計算手法である。気象現象は非線形性が強く、バタフライエフェクトという言葉に象徴されるように初期値などのわずかな違いが結果に大きな相違をもたらす。従って個々のシミュレーションが不可避免的に内在する不確実性を低減し、あるいは不確実性の程度を定量的に評価するためにもアンサンブル計算は不可欠の手法である。h3-Open-UTIL/MP は連成されたモデルのペアを複数起動し、同時に多数の連成計算を実行することができる。もちろんこれだけなら複数のジョブを同時に投入することでも実現可能であるが、h3-Open-UTIL/MP は実行中の連成モデル群に対して、連成間通信を行う機能を有している。通信の模式図を図 13 に示す。図のように h3-Open-UTIL/MP のアンサンブル連成は 2 通りの実行方法をサポートしている。ひとつは左図のように、連成されたペア（例えば大気海洋結合モデル）を複数起動する方法であり、もう一つは右図のように複数のモデル A に対してひとつのモデル B を連成するような方法である。後者の具体的な適用例を図 14 に示す。ここでは低解像度大気モデルのアンサンブルと高解像度大気モデルが連成されている。この手法により、高解像度モデルのアンサンブル計算に比してより少ない計算資源で、確度の高い高解像度計算が実現できる。

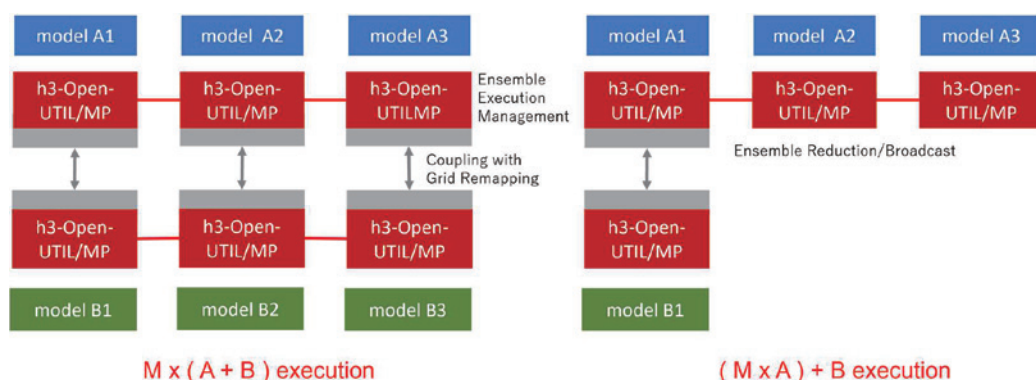


図 13 アンサンブル連成の実行パターン

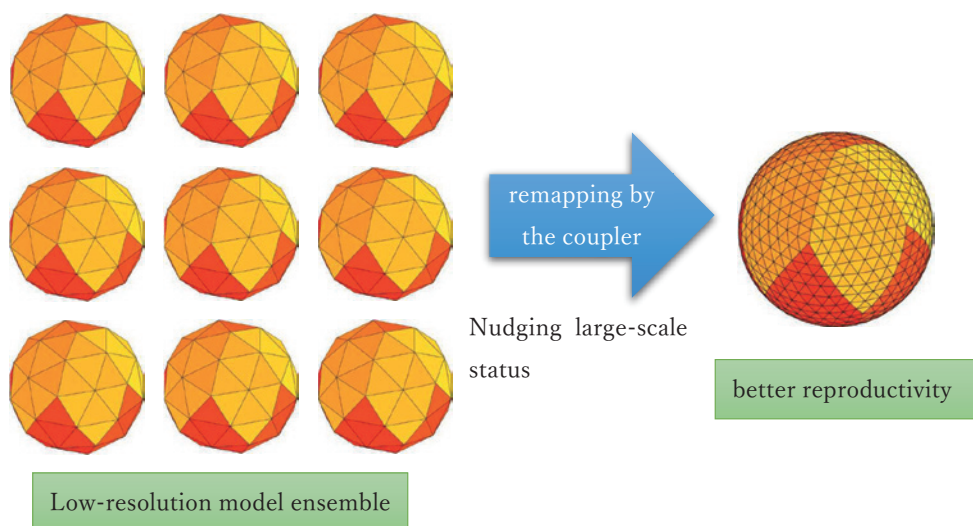


図 14 多対一アンサンブル連成の適用例

4.2 アンサンブル連成の方法

本節では h3-Open-UTIL/MP においてアンサンブル連成がどのように実現されているかを解説する。説明の便のために、ここでアンサンブル方向と連成方向という用語を導入する。アンサンブル方向とは図 13 では横方向になる。アンサンブル方向ではアンサンブル対象となる各モデルコンポーネントは当然ながら同一のモデルであり、更に同じ時空間設定を持つ、すなわち同一の格子、同一の領域分割、同一の積分時間ステップ等を持つものとする。一方連成方向は図 13 では縦方向となり、対象となるモデルコンポーネントは相互に異なる時空間構造を持つ異なるモデル（同一モデルも含む）であることが前提となる。アンサンブル計算におけるアンサンブル数は目的や物理的条件、あるいは利用可能な計算資源量などで異なるが、大規模な計算になると 1000 を越える[4]。このアンサンブル数を MPI で起動する際に `mpirun -np N ensemble1 : -np N ensemble2 : -np N ensemble3` のようにひとつずつ記述していくのは現実的ではない。そこで h3-Open-UTIL/MP では `mpirun -np MxNA ModelA : -np MxNB ModelB` のように各モデルで実行される全プロセス数を `mpi` への引数とする。ここで `M` はアンサンブル数、`NA` は `ModelA` における個々のアンサンブルのプロセス数、`NB` は `ModelB` での個々のアンサンブルのプロセス数である。アンサンブル数 `M` は h3-Open-UTIL/MP の初期化 API である `h3ou_init` で引数として与える。`h3ou_init` はデフォルトでは引数としてモデル名称と設定ファイル名の 2 つを与えるが、3 番目に `optional` 引数としてアンサンブル数(`num_of_ensemble`)を与えることができるようになっている。2 つのモデルで `num_of_ensemble` は同数かもしくは片方が 1 の場合のみが許され、同数の場合は図 13 左図の形式、片方が 1 の場合は図 13 右図の形式と判定される。また各モデルの全プロセス数はアンサンブル数で割り切れることが条件である。`h3ou_init` の引数として与えられるモデルの名称および MPI 環境から得られる全プロセス数から、各モデルが合計いくつのプロセスで実行されるかが求められ、この値とアンサンブル数 `M` からアンサンブル毎のプロセス数 `NA,NB` も求められる。引数として与えるモデル名はカブラ内部ではモデル識別コードとして扱われ、実際のモデル名は `ModelA0001`, `ModelA0002` のようにアンサンブル番号が付加された文字列となる。これが個々のモデル名としてログファイル名などに用いられるようになっている。個々のプロセスが何番のアンサンブルに所属しているかは h3-Open-UTIL/MP の API を通じて取得できるので、アンサンブル毎にデータやパラメータを変える際に、所属番号に応じてデータのディレクトリを変える、あるいは設定ファイル名を変えるなどの対応が可能となる。アンサンブル連成に際して h3-Open-UTIL/MP が生成し使用している MPI コミュニケータの種類を図 15 に示す。最外周の黒線は MPI のデフォルトコミュニケータ `MPI_COMM_WORLD` である。各モデルプロセス全体を囲む青線がアンサンブル方向（モデル毎）のグローバル通信を行うコミュニケータである。個々のモデルを囲む緑線が、各モデルコンポーネントが通信（例えば袖領域の交換）に用いるローカルコミュニケータである。連成方向に 2 つのモデルコンポーネントを囲む黄色線が個々のアンサンブルで連成に用いられるコミュニケータである。最後に、アンサンブル方向でモデル内の同一プロセス同士を繋いでいる赤線がアンサンブル方向のプロセス間通信を行うコミュニケータである。これらのコミュニケータの内、利用者が参照することができ、かつ参照する必要があるのはローカルコミュニケータだけであり、残りのコミュニケータはすべてカブラ内部で使用される。

ンサンプル数を取得して `h3ou_init` の引数として渡している。アンサンプルの情報はログファイルの冒頭付近に出力される。サンプル実行のログを図 16 に示す。サンプル実行は多対一アンサンプルであるため 0 番アンサンプルとそれ以外で動作が異なる。図 16 上図は 0 番アンサンプルの出力、下図は 0 番以外の出力である。0 番は相手モデルとの結合フラグ(coupling flag)が T になっているが、その他のアンサンプルは結合フラグが F となっており直接連成されていないことがわかる。

```

subroutine init_common(comp_name, comp_id)
  use h3ou_api, only : h3ou_init, h3ou_get_mpi_parameter
  implicit none
  character(len=*) , intent(IN) :: comp_name
  integer, intent(IN) :: comp_id
  integer, parameter :: MAX_GRID = 8
  integer :: ierror
  integer :: num_of_arg
  character(len=128) :: arg_ensemble
  integer :: num_of_ensemble
  intrinsic :: command_argument_count, get_command_argument

  my_name = trim(comp_name)
  my_comp_id = comp_id

  num_of_arg = command_argument_count()

  if (num_of_arg == 0) then
    num_of_ensemble = 0
  else
    call get_command_argument(1, arg_ensemble)
    read(trim(arg_ensemble), *) num_of_ensemble
  end if

  if (num_of_ensemble >= 1) then
    call h3ou_init(my_name, "coupling.conf", num_of_ensemble)
  else
    call h3ou_init(my_name, "coupling.conf")
  end if
end if

```

図 16 アンサンブル連成の初期化プログラム

```
>>>>>>>>>>>>>>> h3ou_init IN
----- ensemble information
Ensemble coupling : my name =   app10000, target name =   app20000
ensemble type = 2, coupling flag = T
```

```
>>>>>>>>>>>>> h3ou_init IN
----- ensemble information
Ensemble coupling : my name =   app10001, target name =   NO_TARGET
ensemble type = 2, couplng flag =    F
```

図 17 アンサンブル連成のログ（上図：0 番アンサンブル、下図：0 番以外のアンサンブル）

5 まとめ

本稿では Wisteria/BDEC-01 利用事例として汎用連成ソフトウェア h3-Open-UTIL/MP の拡張機能について解説した。h3-Open-SYS/WaitIO との協調機能を用いることで Odyssey と Aquarius の 2 つのシステム間での連成計算が可能となり、更に Python インターフェースを供用することによってシミュレーションプログラムと機械学習プログラムなど従来はファイルを経由して逐次的に実行されていた処理をリアルタイムに並列処理することができるようになる。本稿では Toy モデルを用いた簡単な計算事例を紹介したが、これらの機能を実アプリケーションに適用した計算も試みられている [5]。今後も機能拡張やユーザインターフェースの改良といったカプラ本体の開発を継続するとともに、実アプリケーションへの適用も推進してゆく予定である。

参考文献

- [1] 荒川隆, 八代尚, 中島研吾, Wisteria/BDEC-01 利用事例(5) マルチプログラム連成ライブラリ h3-Open-UTIL/MP(1/2), スーパーコンピューティングニュース Vol.24 No.3, https://www.cc.u-tokyo.ac.jp/public/VOL24/No3/13_202205-Wisteria-2.pdf
- [2] 住元真司, 坂口吉生, 松葉浩也, 中島研吾, Wisteria/BDEC-01 利用事例(3) データ受け渡しライブラリ h3-Open-SYS/WaitIO(1/2), スーパーコンピューティングニュース Vol.24 No.2, https://www.cc.u-tokyo.ac.jp/public/VOL24/No2/10_202203Wisteria-1.pdf
- [3] 住元真司, 坂口吉生, 松葉浩也, 中島研吾, Wisteria/BDEC-01 利用事例(4) データ受け渡しライブラリ h3-Open-SYS/WaitIO(2/2), スーパーコンピューティングニュース Vol.24 No.3, https://www.cc.u-tokyo.ac.jp/public/VOL24/No3/12_202205-Wisteria-1.pdf
- [4] H. Yashiro, H., Terasaki, K., Kawai, Y., Kudo, S., Miyoshi, T., Imamura, T., Minami, K., Inoue, H., Nishiki, T., Saji, T., Satoh, M., and Tomita, H.: "A 1024- Member Ensemble Data Assimilation with 3.5-Km Mesh Global Weather Simulations," in SC20: International Conference for High Performance Computing, Networking, Storage and Analysis (SC), Atlanta, GA, US, 2020 pp. 1-10. doi: 10.1109/SC41405.2020.00005
- [5] Arakawa, T., Yashiro, H., Nakajima, K., Development of a coupler h3-Open-UTIL/MP, ACM Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region(HPC Asia 2022), 2022, <https://doi.org/10.1145/3492805.3492809>

ミニマルスパン・チャネル乱流の直接数値計算による 乱流伝熱解析

関本 敦

岡山大学学術研究院 環境生命科学学域

1. はじめに

壁に沿う流れは工学上極めて重要で、壁面近傍での乱流現象は熱交換器などの壁面摩擦抵抗や熱伝達に大きく影響する。壁乱流の予測制御のためには乱流統計量のデータ取得が必要である。実験計測ではすべての統計量を精度良く計測することは困難であるため、もっぱら支配方程式である Navier-Stokes 方程式をモデルを使わずに直接解く数値シミュレーション (DNS) が用いられる。乱流の最小スケール渦までを解像できる DNS は、実測では計測困難な三次元の流れ場の情報が得られるため、乱流現象や統計法則の解明および予測モデルの構築には必要不可欠なツールである。また、得られたデータは、三次元空間内の時系列データというビッグデータであり、その解析には大規模な計算環境や数値解析の技術が要求される。時々刻々と変化する乱流のダイナミクスの中には、レアなイベントが含まれるため、乱流統計量を正確に取得するためには、長時間の DNS 計算や複数の初期データから多くの試行計算が必要であり、HPC 環境の発展と共に研究も進展している。

壁近傍の乱流構造は、統計量の予測のみならず、制御対象としても注目すべき対象であり、微細渦から大規模構造まで階層構造を有している。壁面からの熱移動の主要因となる階層化された渦の相互作用の特徴を抽出することが、今後の高精度な乱流モデルや流れの制御、高効率な伝熱機器の設計開発の鍵となる。本研究で対象としたのは、水のような中程度のプラントル数流体から、化学プロセスにみられる高プラントル数流体、もしくは海洋 (海水) などの高シュミット数流体である。温度の移流拡散方程式と濃度の移流拡散方程式は本質的に同じ形で、区別する必要はないので、本研究では、伝熱問題を対象として、温度成層がある壁面近傍の乱流を考える。プラントル数 (Pr) は粘性係数と熱拡散率との比を表し、高プラントル数流体では、運動量の拡散よりも温度拡散の方が遅くなる。したがって、乱流中の最小スケールの渦に比べて、 $Pr^{1/2}$ 倍で小さなスケールの温度変動が生じるため、非常に多くの計算格子が必要となり、高プラントル数乱流の DNS は現在でも挑戦的な課題である。最近の一樣等方性乱流を対象とした研究では、速度場も温度場と同様に高解像度を用いて、浮力が微小スケールに及ぼす影響も詳細に数値シミュレーションで再現されている [1, 2]。壁面近傍の流れにおいては、高プラントル数の DNS を扱う例はまだ少なく、 $Pr = 100$ 程度の壁乱流の詳細を十分な解像で捉えた研究はまだ少ない。特に、壁面付近の熱伝達は乱流現象の影響が大きく、浮力による乱流中の大小様々な渦への浮力の影響を詳細に調べることは、乱流モデルのさらなる高精度化への基礎的な知見となる。

本研究では、乱流の最小渦と壁面の相互作用下における、温度場の最小スケール構造の振る舞いを詳細に調べるために、計算領域をスパン方向に絞ることで、速度の最小スケールと温度の最小スケールまでを解像した直接数値シミュレーションを行う。計算コストを抑えながら、浮力の強さのパラメータであるフルード数 Fr を変えた DNS 計算を実施できる。十分高いレイノルズ数であれば、このような計算手法によっても、壁面近傍の乱流状態を正しく捉えることができ、同様の手法

は粗面乱流の研究でも用いられている [3, 4, 5]. 近年, 工学現場でも利用されつつあるラージ・エディ・シミュレーション (LES) においても, 乱流プラントル数という経験定数が扱われるが, その壁面近傍の分布と他のパラメータとの関連性は未解明である. また, 近年では機械学習による超解像や, ニューラルネットワークによる物理モデリングなどのための乱流の壁面近傍モデルの高精度化に必要な不可欠なビッグデータとして有効利用できる.

2. 結果

壁面近傍の乱流伝熱もしくは物質拡散の予測精度向上に向けて, 非圧縮性流体の平行平板間乱流の高精度直接数値計算 (DNS) を実施した. 壁乱流の DNS には, 少ない格子点数でも高精度に計算できるスペクトル法を用いており, 計算手法については [5] で述べられているように, Legendre-Chebyshev 変換を用いて, 高精度に Helmholtz 方程式を解く. 乱流渦を十分に解像するために壁面近傍に格子を細かく切るため, 時間ステップの制約が大きくなる. これを緩和するために, 粘性項は時間に対して半陰的に取り扱う. 温度場による浮力効果については, 浮力項のみに密度変化を考慮するブシネスク近似を用いる. OakForest-PACS を利用して MPI と OpenMP のハイブリッド環境で計算を実行し, 壁面熱伝達へのプラントル数の影響を調査するため, $Pr = 1$ 程度から 100 までの DNS データを取得した.

高プラントル数流体においては, $Pr^{1/2}$ で温度境界層が薄くなるため, 例えば, $Pr = 100$ (油など) では空気 ($Pr = 0.7$) と比べて, 1 方向あたり 10 倍の格子解像度が必要である. 本課題ではスパン方向に計算領域を狭めた計算によって, 高プラントル数の壁面乱流伝熱の DNS を比較的低コストで実現できた. 乱流統計量には, 壁面近傍の速度変動のピーク位置よりも更に壁に近い位置で, 温度変動のピークが観測でき, 壁面近傍の温度変動の散逸スケールを十分に捉えることができることを確認した.

図 1 では, 非常に薄い温度の層が再現できている. 壁面近傍では, 温度層が何層にも折り畳まれることで, 伝熱が促進される. この層は流れ方向に非常に引き延ばされながら, 壁面近傍の乱流渦と相互作用していると考えられる. 壁面近傍の温度スカラー場の折りたたみ現象については, これまでに詳細な研究報告はなく, 壁面の温度変動や熱伝達率への寄与について詳細な解析が必要であるが, 比較的簡単な熱輸送機構であれば, モデル化することによって, 高プラントル数流体の伝熱シミュレーションの計算不可の大幅な低減に繋がるものと期待できる. また, この折りたたみ熱伝達機構の制御を目指せば, 高効率な乱流伝熱を実現できる可能性も広がる. 今後, 本課題で得られた DNS の時系列データベースを用いて, 詳細な伝熱ダイナミクスの解明を進める.

謝辞

本研究での計算は, 東京大学情報基盤センター若手利用 (2019 年度) の援助を受け, OakFoorest-PACKS を利用した.

参考文献

- [1] S. Okino, H. Hanazaki, *J. Fluid Mech.* **874**, 821 (2019).
- [2] S. Okino, H. Hanazaki, *J. Fluid Mech.* **891**, A19 (2020).
- [3] O. Flores, J. Jiménez, *Phys. Fluids* **22**, 071704 (2010).
- [4] M. MacDonald, *et al.*, *J. Fluid Mech.* **816**, 5 (2017).

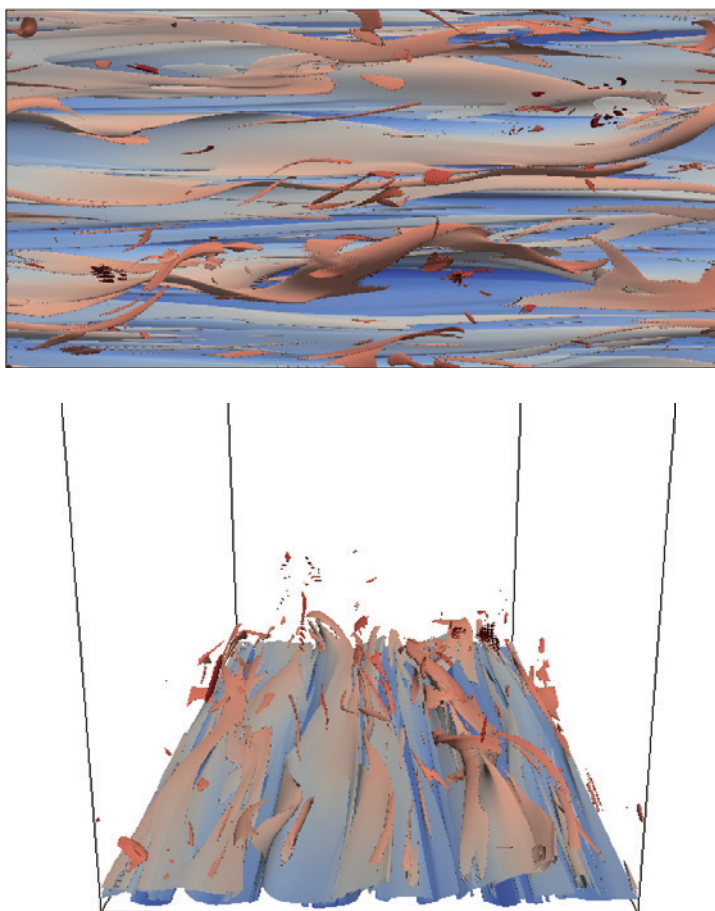


図1 $Re_\tau = 350$, $Pr = 100$ のときの冷却壁面近傍の温度の等値面 ($T/\Delta T = -0.25$) で色は流れ方向の速さを示す (赤が高速, 青が低速部). (上図) Top view (流れは左から右). (下図) 3D view (流れは手前から奥).

- [5] A. Sekimoto, C. Atkinson, J. Soria, *J. Phys.: Conf. Series* **1001**, 012020 (2018).

Geomechanical modeling for subsurface CO₂ storage with active pressure management: A preliminary study

Yi Zhang

1. Geological Carbon Dioxide Storage Technology Research Association
2. Research Institute of Innovative Technology for the Earth

1. Introduction

Geological CO₂ storage (GCS) in subsurface aquifers has been proposed as a near-term solution for reducing CO₂ emission and mitigating climate change. The concept of CGS – injection of CO₂ captured from industrial processes into rock formations deep underground – is quite straightforward to understand. However, the projects with large-scale CO₂ injection may face challenges. The pressure buildup in the aquifer due to large-scale CO₂ injection may induce geomechanical issues, such as fault reactivation, induced seismicity, fluid leakage and surface uplift. The occurrence of these issues is thought to be negative to the project operation and public acceptance of GCS. Therefore, monitoring and modeling of geomechanical behavior of reservoir-caprock system are quite critical for a successful GCS project.

To mitigating pressure buildup and geomechanical risks, one approach called Active reservoir Pressure Management (APM) conducted by production reservoir brine has been proposed (Buscheck et al., 2011). The production of brine from reservoir can let the reservoir pressure (buildup) be partially relaxed and leaves more pore space to CO₂. Some of the benefits of APM include reduced geomechanical risks and increased CO₂ storage capacity. On the other hand, the brine production operation can also increase the cost.

In practice, when applying the APM approach in a project, the operator needs in situ information to understand the state of pressure buildup and geomechanical deformation and further to determine the rate of CO₂ injection and brine production. Recently, distributed strain sensing (DSS) using optical fiber has been demonstrated as a valid monitoring method to provide such information. The DSS method can offer real-time and continuous profiles of in situ vertical strain along the fiber cable installed wellbore. One task left for APM is to understand the signal of measured strain response when performing the injection and production. A better understanding of the strain response helps for an optimized management of injection and production procedures, controlling the potential geomechanical risks and supporting the decision making.

In this study, we conducted geomechanical numerical modeling for APM with an emphasis of vertical strain changes during CO₂ injection with and without brine production. We performed the modeling job on Oakbridge-CX Supercomputer System.

2. Mathematical equations

The modeling involves solving coupled equations of mass and flux of two-phase fluid flow and solid mechanics as follows.

(1) The mass conservation for fluid species κ is described by the equation

$$\frac{\partial M^\kappa}{\partial t} + \nabla \cdot \mathbf{F}^\kappa = q^\kappa$$

where M^κ is the mass, t is the time, q^κ is the source and $\mathbf{F}^\kappa$ is the advective flux, which is governed by the extended Darcy's law.

(2) The mechanical equation is expressed by stress with the coupling between effective stress and pore pressure through the Biot coefficient.

$$\sigma_{ij}^{\text{eff}} = \sigma_{ij}^{\text{tot}} + \alpha_B P_f$$

where σ^{eff} is the effective stress tensor, σ^{tot} is the total stress tensor, P_f is the pore pressure, and α_B is the Biot coefficient. The strain $\epsilon_{kl}^{\text{elastic}}$ is associated with the effective stress through the solid-mechanical constitutive law:

$$\sigma_{ij}^{\text{eff}} = E_{ijkl} \epsilon_{kl}^{\text{elastic}}$$

where E_{ijkl} is the elastic tensor.

3. Numerical modeling

The open-source code MOOSE is used for solving the two-phase fluid flow geomechanical model. MOOSE is a multiphysics FEM framework with the PETSc non-linear solver package and libmesh to provide the finite element discretization. The mumps SMP preconditioner and Newton solver with automatic scaling are applied.

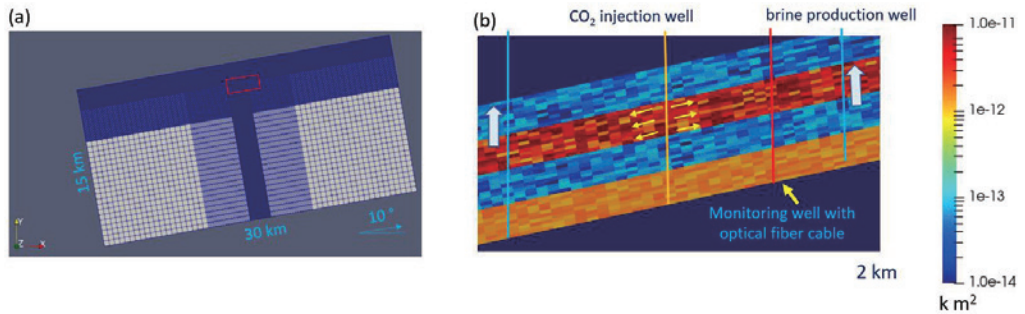


Figure 1. The numerical model. (a) The full domain model with the meshing; (b) The local region near CO2 injection well and brine production well.

In this preliminary study, for simplicity, we perform the modeling using a dipping two-dimensional model (so having dimensional effect compared to a three-dimensional model, Figure 1). The model is intentionally set by several units with different permeability ($1\text{e-}14 \sim 1\text{e-}11 \text{ m}^2$). In each layer, Gaussian distributed heterogeneities

are added. Otherwise, uniform elastic constants (bulk modulus 6 GPa, Poisson' ratio 0.2 and Biot' s coefficient 1.0) are used. To avoid the boundary effect, we set a much larger modeling domain (30 km×15 km). We use dense Cartesian mesh gridding within the ROI and near-by regions and sparse gridding at other regions. A constant CO₂ injection rate (20t/day) is set. Cases with and without brine production are considered. The rate of brine production is the same as CO₂ injection rate.

4. Results

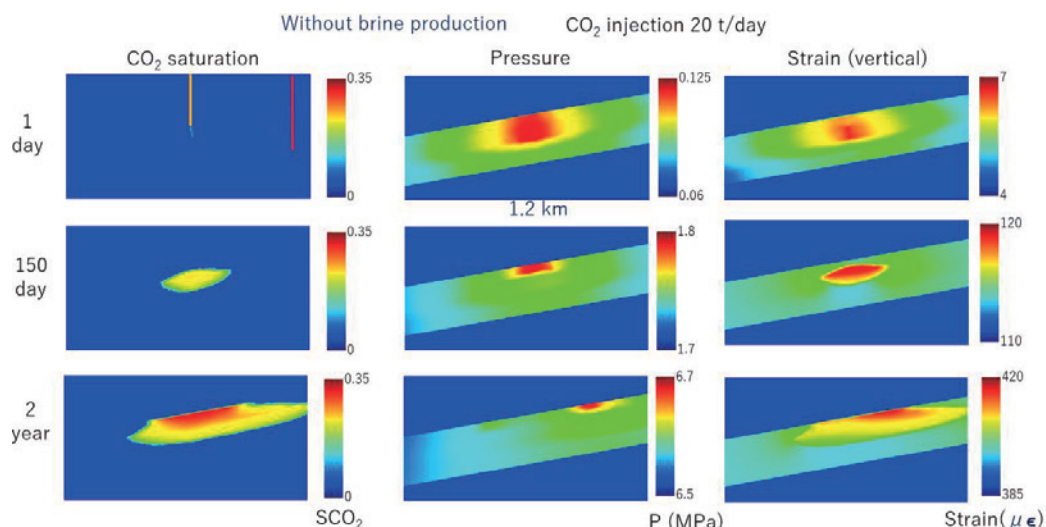


Figure 2. The results numerical modeling for the case without brine production.

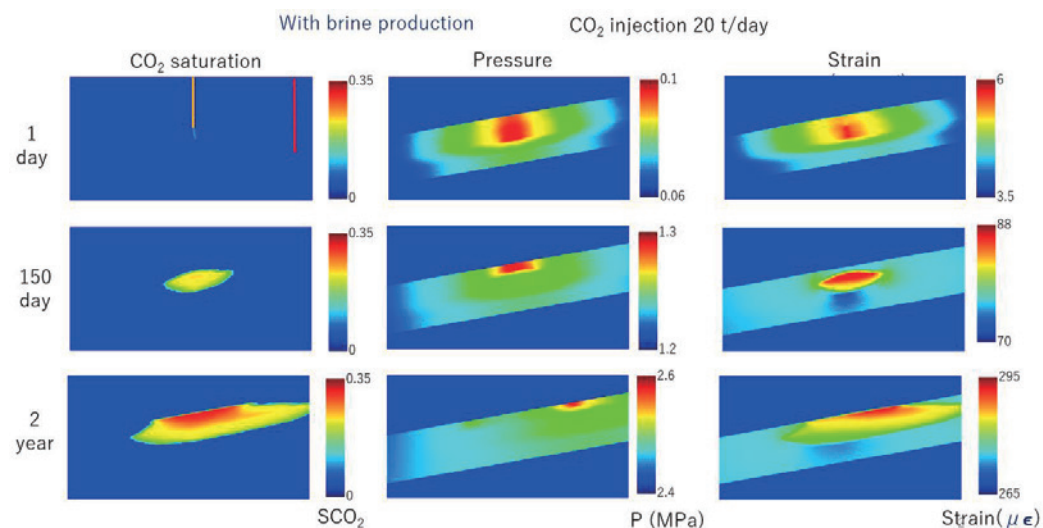


Figure 3. The results numerical modeling for the case with brine production.

Figures 2 and 3 show the modeling results modeled CO₂ saturation, pressure, and vertical strain at different time since CO₂ injection for the two cases without and

with brine injection. We see that the CO₂ plume distribution for the two cases is quite similar at the injection and production flow rate. In contrast, the shapes and magnitudes of fluid pressure distribution and strain between them are quite different. With the brine production, the magnitude of pressure changes and deformation is reduced as expected. This supports the effectiveness of APM through brine production and using DSS as a monitoring tool. Moreover, we can find that dilation strain developed around the injection well. At the earlier stage (since the injection beginning), the strain pattern is mainly induced by the pore pressure diffusion and the strain magnitude is small ($< 10 \mu\epsilon$). There is an obvious strain front corresponding to the pressure front. Later, the strain pattern is affected by the development of CO₂ plume. This is because the relatively large pressure changes near the plume boundary due to the changes in relative permeability and capillary pressure. Therefore, in addition to the geomechanical purpose, we can use the strain signal to understand the state of pressure and CO₂ plume migration.

5. Conclusions

In this short report, we presented our preliminary modeling results of geomechanical modeling of CO₂ injection with active pressure management in subsurface aquifer. We emphasized the deployment of distributed strain sensing method for the monitoring purpose. The modeling results suggest that the strain measurement can be used for a better understanding of the state of pressure buildup and geomechanical risks and perhaps CO₂ plume migration. Therefore, DSS can be expected to play an important role in CO₂ storage project. More investigations are needed to evaluate the effects of hydromechanical parameters, well placement, brine extraction interval and rate.

Acknowledgements

This paper is based on results obtained from a project (JPNP18006) commissioned by the New Energy and Industrial Technology Development Organization (NEDO) and the Ministry of Economy, Trade and Industry (METI) of Japan. This research is partially supported by Initiative on Promotion of Supercomputing for Young or Women Researchers, Information Technology Center, The University of Tokyo.

Efficient Simulation of Ferromagnetic Spin Configuration

李其放 (LI Qifang)

東京大学物性研究所

1. Introduction to shape anisotropy

In ferromagnetic materials, spins of atoms prefer to align in a parallel way below the Curie temperature T_c , as a result of the ferromagnetic nearest-neighboring interaction. Classical Monte Carlo algorithm is a powerful and efficient tool to simulate the spin configuration of ferromagnets by using Heisenberg model. However, realistic ferromagnets usually involve many other interactions, like dipole-dipole interaction (DDI), crystalline anisotropy, Zeeman energy, etc.

Considering the boundary effect of ferromagnets, shape anisotropy will affect the spin configuration. This anisotropy comes from the long-ranged DDI. As a result in long thin ferromagnets, spins prefer to align parallel to the long side of ferromagnets. In addition, magnetic domains of flux-closure shape can be observed in ferromagnets, as the length and width of rectangle are approximately equivalent. These two kinds of spin configuration is related to not only the competition of the nearest-neighboring exchange interaction J and dipole-dipole interaction D , but also the shape of the ferromagnets. Shape anisotropy plays an important

We simulate the spin configuration from a first-principle perspective and study the phase diagram of spin configuration with different system sizes and interaction constants.

All these interactions, except DDI, are short-ranged interactions, so the time complexity of simulating is in order of $O(N)$ per Monte Carlo steps (MCS). Here, N is the number of independent spins. Unfortunately, the long-ranged DDI exists between every two spins, so the time complexity is in order of $O(N^2)$ per Monte Carlo steps (MCS).

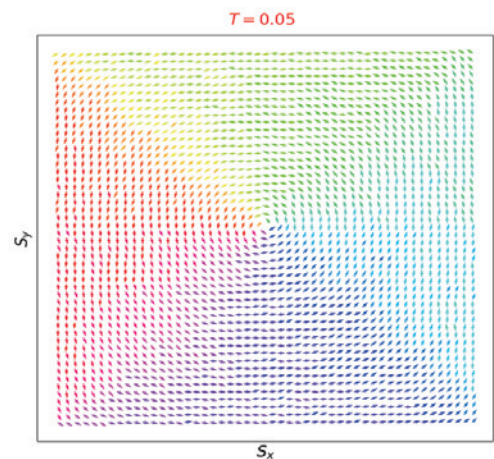


Fig. 1 Vortex spin configuration in 50×50 spins systems ($J = 1$, $D = 0.1$)

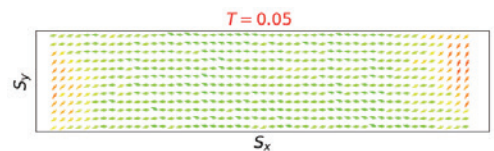


Fig. 2 Spin configuration parallel to long side in 50×12 spins systems ($J = 1$, $D = 0.1$)

2. Efficient Simulation with Stochastic Cutoff (SCO) algorithm

In Monte Carlo simulation for magnetic systems, the great majority of the DDI terms is almost negligible, comparing to the exchange interaction, so they have hardly any influence on the accept-or-reject decision for new samplings. SCO algorithm utilizes this feature and switch most of the unimportant DDI terms to 0, and the others to a different $\bar{V}$ to satisfy the detailed balance condition.

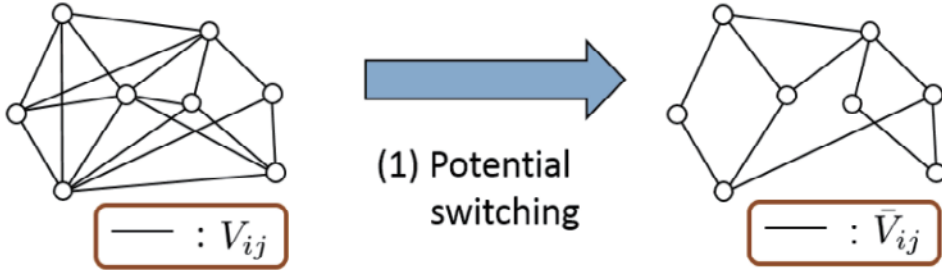


Fig. 3 SCO sets the majority of bonds to 0, whereas the values of remaining bonds are switched from V to $\bar{V}$. It is proven that the detailed balance condition is unchanged after the stochastic cutoff.

It has been tested and measured that SCO algorithm can speed-up the Monte Carlo simulation for large-scale ferromagnetic systems.

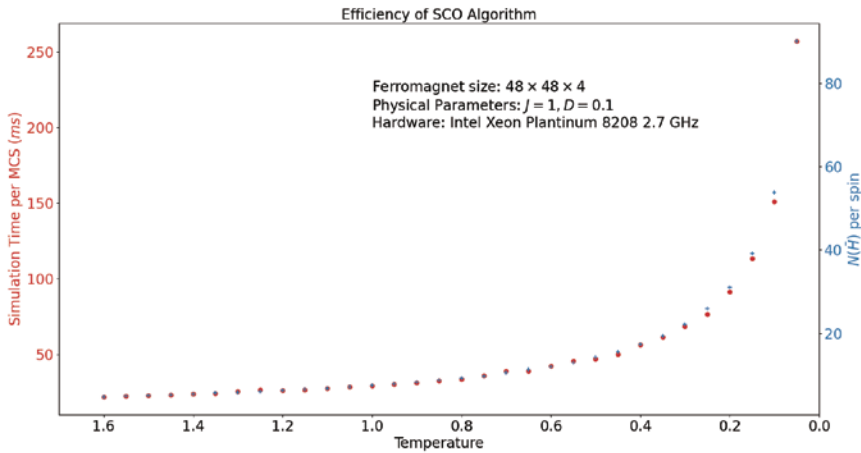


Fig. 4 Efficiency of SCO on Oakbridge-CX for $48 \times 48 \times 4$ ferromagnets. Following the decrease of temperature, more DDI terms are switched to non-zero $\bar{V}$, and the simulation time increases accordingly.

Reference

“Stochastic potential switching algorithm for Monte Carlo simulations of complex systems” *J. Chem. Phys.* **122**, 213110 (2005)

“Order-N cluster Monte Carlo method for spin systems with long-range interactions” *J. Comput. Phys.* **228**, 2629–2642 (2009)

第 176 回お試しアカウント付き並列プログラミング講習会

「MPI 基礎：並列プログラミング入門」実施報告

三木 洋平

東京大学情報基盤センター

2022 年 4 月 26 日（火）、第 176 回お試しアカウント付き並列プログラミング講習会「MPI 基礎：並列プログラミング入門」が開催されました。例年は東京大学情報基盤センターにおいて開催されていた本講習会ですが、今回も新型コロナウイルス感染症対策のために Zoom および Slack を用いたオンライン講習会として実施されました。

本講習会は、東京大学内および学外における当センターのスーパーコンピュータの利用を考えているユーザに加え、社会貢献の一環として、高性能計算や並列処理の技術習得を目的にした企業に所属する研究者、技術者の方が参加可能になっております。

受講者の内訳は、学部学生：2 名、大学院学生：5 名、大学・研究機関教職員：6 名、企業：4 名で、合計 17 名でした。

1 ヶ月間有効となるお試しアカウントが与えられ、Wisteria/BDEC-01 スーパーコンピュータシステムの利用方法、MPI（Message Passing Interface）を用いたプログラミングに関する基礎演習が、1 日終日の日程で行われました。

当日のプログラムを、以下に載せます。

- 4 月 26 日（火）
 - 10：00 - 11：20 テストプログラムの実行など（演習）
 - 11：30 - 12：30 並列プログラミングの基本（座学）
 - 13：40 - 14：40 MPI プログラム実習Ⅰ（演習）
 - 14：50 - 15：50 MPI プログラミング実習Ⅱ（演習）
 - 16：00 - 17：00 MPI プログラミング実習Ⅲ（演習）

対面での講習会として開催してきた本講習会では例年スパコンへのログイン支援を当日行ってきましたが、新型コロナウイルス感染症対策のためにオンライン開催ということもあり、事前に配布した資料を参考に事前にログインしてもらうこととしました。これは去年度の講習会と同様の対応です。また、座学パートについては Zoom 講習会を録画したものを、東京大学情報基盤センタースーパーコンピューティング部門の YouTube チャンネル (<https://www.youtube.com/channel/UC2CHaGp1A0-vqR1V7wmU0-w/videos>) にアップロードした（座学：<https://youtu.be/sLJ8wFFE-QM>）のでいつでも視聴できるようになっています。

17 名中 11 名の参加者について、講習会に関するアンケートをご提出いただきました。

主要な項目の集計結果を以下に掲載します。

プログラミング経験については0年の方から、最長で35年と回答された方もおり幅広く分布していました（図1）。並列プログラミング経験については、ある方が6名、まったくない方が5名と拮抗していました。並列プログラミング経験があると答えられた方は数年程度の方がほとんどでしたが、例年の参加者に比べてプログラミング経験・並列プログラミング経験の豊富な方が多かったようです。普段使用しているプログラミング言語について

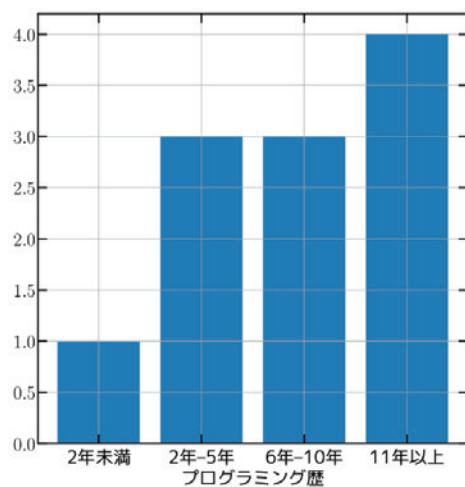


図1：参加者のプログラミング経験

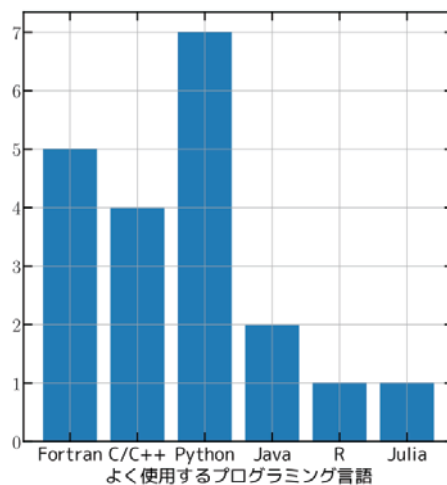


図2：普段使用しているプログラミング言語

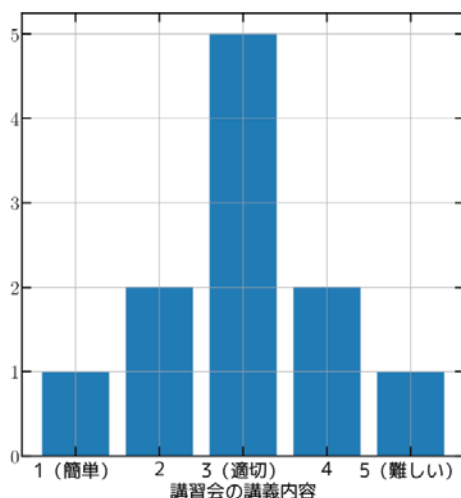


図3：講習会の講義内容（プレゼン）

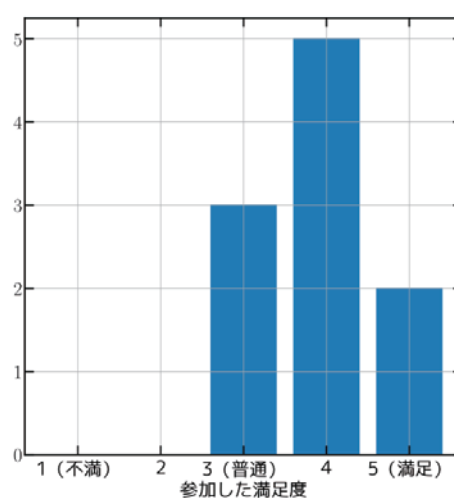


図4：講習会に参加した満足度

はPythonが最も多く、C/C++及びFortranはほぼ同数でした（図2）。

講習会の難易度設定については、適切であると回答された参加者が最多であり、また簡単

あるいは難しいと回答された方が均等に分布しており（図 3）、妥当な難易度設定と考えられます。また、講習会に参加した満足度は図 4 に示した通りであり、おおむね満足度が高かったようです。

今回も Zoom を用いての完全オンライン開催であったので、オンライン開催に関する回答をいただきました。オンライン開催で良かったことについての回答は

- 用語などわかっていないことが多いことを確認することができた点。
- 説明がわかりやすく、楽しく受講できた。
- 今まで知っていた部分の再確認もできてよかった。
- 双方向形式で指導して頂ける講習会に、手軽に参加できる点。
- slack との併用をしていただいたため、質問をしやすい環境にあった点
- slack を平行して利用していたこと
- 音声でコミュニケーションできるのは、質問時にこちらの理解を講師の先生が把握して補足していただける点でよいと思いました。
- 自分のペースでできるところ

というものでした。Zoom および slack を用いて双方向的に進められたという点の評価が高かったです。また、自由記述欄においては以下の感想をいただきました。

- python を数か月前から始めた私でもサンプルプログラムの 4 番まではだんだんといいていくことができたので、難易度はちょうどよいと思います。要望としましては、MPI_Send などの関数の引数についてもう少し説明をしていただければ、何を与える関数を動かすために、何を代入しているのかという理解が深まると思いました。
- 以前スパコンを触ったときには各システム特有の事情（独自のコマンド等）とスパコン一般の事情（セキュリティやジョブスケジューラ等）が切り分けられずに苦勞したのですが、この講習会の資料ではそのあたりが丁寧に書かれており理解がスムーズに進みました。
- 並列化有限要素法や線形解法の並列化等のテーマでの開催があれば喜んで参加します。
- 操作を実演しながら、プログラムなどの説明を行ってもらいたかった。
- 拡散方程式のサンプルプログラムの詳細な説明を付録のような形で頂けるとありがたいです。サンプルプログラムの流れを理解するのに時間がかかりました。
- やはり後半の二次元の拡散方程式に関する内容が圧倒的に難しく感じました。
- 前半の総和演算に関しては何となく理解できたものの、講義内容と演習内容を関連付けて理解できなかったのが残念でした。
- 本日の講義終了後、時間がある時に講義資料やソースコードを振り返って、一つずつ丁寧に理解していきたいと思います。
- 勉強したい点が上級編にありそうなので、上級編もチャレンジしてみます

同様の講習会があれば、「また受けたい」という回答が9名、「どちらともいえない」が2名であり、その他の講習会にも期待されていることが伺えます。

以上

第 177 回お試しアカウント付き並列プログラミング講習会 「MATLAB の実行方法」実施報告

東京大学情報基盤センター

本稿では、2022 年 4 月 27 日（水）にオンライン開催した第 177 回お試しアカウント付き並列プログラミング講習会「MATLAB の実行方法¹⁾」（共催：東京大学情報基盤センター、MathWorks Japan、PC クラスタコンソーシアム（実用アプリケーション部会・HPC オープンソースソフトウェア普及部会））について報告する。

東京大学情報基盤センター（以下、当センター）では 2007 年からスーパーコンピュータを使用した「お試しアカウント付き並列プログラミング講習会」を開催しているが、新型コロナウイルス感染症対策のため、2020 年 4 月からは全ての講習会を Zoom 使用による「オンライン」講習会として実施している。

MATLAB²⁾ は MathWorks 社が提供する数値解析ソフトウェアである。当センターでは、MATLAB におけるスーパーコンピュータ利用者の自作あるいはオープンソースのプログラムから関数として呼び出す機能を特に重要と考え、アカデミックユーザ（国公立大学・高専等の教育機関に所属する教職員・学生・研究生等、学位取得プログラムのある機関の教職員等）を対象として、2022 年 2 月 25 日より Wisteria/BDEC-01 Aquarius³⁾ と Oakbridge-CX⁴⁾ において MATLAB の提供を開始した⁵⁾。データ解析、機械学習系に関連する MATLAB の豊富な機能は、利用者の計算科学シミュレーションコードを高度化し、研究の幅を広げることにも貢献するものと期待している。

MATLAB の提供開始を受けて、MATLAB 概要と高速化手法の講義に加えて Wisteria/BDEC-01 Aquarius において MATLAB を実行する演習を行う講習会を 1 日で実施した。本講習会のプログラムを表 1 に示す。

表 1 本講習会のプログラム

第一部 MATLAB 入門とその高速化入門	
13 : 00 - 14 : 00	MATLAB 概要と高速化入門 - MATLAB 概要 - 高速化のためのコードの書き方
第二部 スパコンで MATLAB を動かすハンズオン実習	
14 : 00 - 15 : 00	Wisteria/BDEC-01 概要とログイン
15 : 00 - 15 : 30	休憩
15 : 30 - 17 : 00	MATLAB を起動してサンプルデータを用いた演習 - スーパーコンピュータでの MATLAB 実演 - GPU を使った処理の実演 - Fortran, C のプログラムから MATLAB を呼び出す実演

¹⁾ <https://www.cc.u-tokyo.ac.jp/events/lectures/177/>

²⁾ <https://jp.mathworks.com/products/matlab.html>

³⁾ <https://www.cc.u-tokyo.ac.jp/supercomputer/wisteria/service/>

⁴⁾ <https://www.cc.u-tokyo.ac.jp/supercomputer/obcx/service/>

⁵⁾ <https://www.cc.u-tokyo.ac.jp/guide/application/introduction-matlab.php>

MATLAB を取り扱う講習会は初めての試みであるが、申込締切を前に定員 30 名の申込があり、当日の出席者も 27 名（大学・研究機関教職員 14 名、大学院学生 12 名、その他 1 名）と出席率が高いことから、利用者の MATLAB に対する関心の高さがうかがえる。

講習会終了後にアンケートを実施し、27 名中 21 名の出席者から回答を得ることができた。質問項目と回答（5 段階評価）の人数分布を表 2 に示す。平均満足度は 3.95 であった。

表 2 アンケート集計結果

	評点	1	2	3	4	5
(a) 講習会時間	短い⇔長い		1	14	3	3
(b) 講習会講義内容（プレゼン）	簡単⇔難	1	6	9	4	1
(c) 配布資料内容	簡単⇔難	2	5	10	4	
(d) サンプルプログラム内容	簡単⇔難	1	3	11	5	1
(e) 満足度（平均 3.95）	不満⇔満足		2	3	10	6

アンケートの自由記述欄のコメントを以下に示す。

- Fortran や C から MATLAB を呼べるなど、知らない知識があったので、講習会に参加して認識を新たに出来た。
- テーマによっては大変参考になるので今後も継続実施して頂きたいです。参加者に依るとは思いますがマシンの説明等は既知のことが多いので、もう少し簡略化して頂いても良いかと思います。
- 非常に有益な講習会だと考えているので、時間が合わなかった場合や見返すための録画があれば良いと思いました。
- 本講習会もまた録画にして後日誰でも視聴できるようにする方が良いと思う。
- MATLAB での機械学習やディープラーニング等、応用分野からの切り口で事例紹介（座学）＋ハンズオン形式の講習会があると良いと思います。

第 178 回お試しアカウント付き並列プログラミング講習

「GPU プログラミング入門」

星野哲也

東京大学情報基盤センター

本稿は、2022 年 5 月 18 日にオンライン会議システム Zoom を用いて開催された『第 178 回お試しアカウント付き並列プログラミング講習会：GPU プログラミング入門¹』の開催報告である。

本講習会では、GPU を用いるために必須である、GPU の特性やプログラミング手法に関する講義および実習を行う。対象 GPU としては主に NVIDIA 社の Tesla GPU を、GPU 向けの並列化プログラミング環境としては主に OpenACC を用いる。実習では拡散方程式のカーネルなど、HPC において利用される基本的な計算問題を題材として、GPU 向けのプログラムを作成する方法や、GPU プログラミングを行う上で役に立つツールの使用方法、CPU-GPU 間のデータ転送の最適化を適用するまでの手順を学ぶ。本講習会は 2021 年 5 月 14 日より運用を開始した Wisteria/BDEC-01 の Aquarius ノードを用いて実習を行なった。Aquarius ノードには 8 基の A100 GPU が搭載されているが、講習会受講者が利用できるのは 4 基までである。受講者には 1 ヶ月有効なアカウントが与えられ、1 ノード 30 分までのジョブが実行可能である。講義内容の詳細については、ウェブページ¹から資料をダウンロードできるのでそちらを参照いただきたい。また講習会に用いたサンプルコードは https://github.com/hoshino-UTokyo/lecture_openacc にて公開されている。

合計 17 名の事前登録者があり、13 名（学生：5 名、研究機関：5 名、企業：2 名、その他：1 名）が受講した。講習会終了後にアンケートを実施した（11 名より回収）。表 1 は質問項目と回答（5 段階評価）の人数分布である。

表 1 アンケート集計結果

	評点	1	2	3	4	5
(a) 講習会時間	短い⇔長い			8	2	1
(b) 講習会講義内容（プレゼン）	簡単⇔難		1	7	2	1
(c) 配布資料内容	簡単⇔難	1	1	8	1	
(d) サンプルプログラム内容	簡単⇔難	1		6	3	1
(e) 満足度	不満⇔満足				5	6

¹ <https://www.cc.u-tokyo.ac.jp/events/lectures/>

第 179 回お試しアカウント付き並列プログラミング講習会 OpenMP によるマルチコア・メニィコア並列プログラミング入門 (Wisteria/BDEC-01 (Odyssey, A64FX 搭載)) (オンライン)

中島 研 吾

東京大学情報基盤センター

本稿では、2022 年 5 月 23 日 (月) にオンライン開催した第 179 回お試しアカウント付き並列プログラミング講習会「OpenMP によるマルチコア・メニィコア並列プログラミング入門 (Wisteria/BDEC-01 (Odyssey, A64FX 搭載))」¹⁾ (共催：東京大学情報基盤センター、PC クラスタコンソーシアム (実用アプリケーション部会・HPC オープンソースソフトウェア普及部会)) について紹介する。

東京大学情報基盤センター (以下、本センター) では 2007 年からスーパーコンピュータを使用した「お試しアカウント付き並列プログラミング講習会」を開催しているが、新型コロナウイルス感染症対策のため、2020 年 4 月からは全ての講習会を Zoom 使用による「オンライン」講習会として実施している。「OpenMP によるマルチコア・メニィコア並列プログラミング入門」は、オンラインとなってから、第 131 回 (2020 年 4 月 27 日)、第 143 回 (2020 年 11 月 2 日)、第 151 回 (2021 年 4 月 21 日) と 3 回、Oakbridge-CX (OBCX)²⁾ を使用して実施してきた。2019 年度までは 2 日にわけて実施していたものを、2020 年度からは 1 日にまとめて実施している。

近年マイクロプロセッサのマルチコア化が進み、様々なプログラミングモデルが提案されている。中でも OpenMP は指示行 (ディレクティブ) を挿入するだけで手軽に「並列化」ができるため、広く使用されており、様々な解説書も出版されている。メモリへの書き込みと参照が同時に起こるような「データ依存性 (data dependency)」が生じる場合に並列化を実施するには、適切なデータの並べ替えを施す必要があるが、このような対策は OpenMP 向けの解説書でも詳しく取り上げられることは余り無い。第 151 回以前の講習会では、「有限積法から導かれる疎行列を対象とした ICCG 法」を題材として、科学技術計算のためのマルチコアプログラミングにおいて重要なデータ配置、reordering などのアルゴリズムについての講義、スパコンを使用した実習を実施した。内容は筆者が本学大学院工学系研究科・情報理工学系研究科の講義として実施している「計算科学アライアンス特別講義 I」、「科学技術計算 I」、「スレッド並列コンピューティング」の内容に準拠している³⁾。

2021 年度からは、2021 年 5 月 14 日に運用を開始した Wisteria/BDEC-01 (Odyssey)⁴⁾ 導入に合わせて内容を一新し、データ依存性に関連した内容を完全に削除したカリキュラムを策定した。大体 105 分の講義にして 4 回程度の内容であり、1 日、450 分程度で実施するには手頃な量となっている。

運用開始から 1 年近くが経過し、2021 年 5 月の時点では性能にやや問題があった C コンパイラの性能等も改善された。また性能評価のための詳細プロファイラの使用法など新たな項目を

¹⁾ <https://www.cc.u-tokyo.ac.jp/events/lectures/179/>

²⁾ <https://www.cc.u-tokyo.ac.jp/supercomputer/obcx/service/>

³⁾ <http://nkl.cc.u-tokyo.ac.jp/21s/>

⁴⁾ <https://www.cc.u-tokyo.ac.jp/supercomputer/wisteria/service/>

追加するなど、内容としては前回と比較して充実したものとなった。

合計 9 名の参加申込みがあり、受講者は 7 名（大学・研究機関教職員：1，大学院生：2，企業：4）であった。アンケート結果（回収 7 名）について表 2 に示す。平均満足度は 3.86 と高めであった。満足度を「3」としたうちの 1 名は、表 2 の（a）～（d）が全て「5」であった。

表 1 本講習会の概要

09：00-11：00	有限体積法
11：00-12：30	OpenMP 入門
13：30-16：00	オーダリング
16：00-17：30	OpenMP 並列化
17：30-18：00	質疑

表 2 アンケート集計結果

	評点	1	2	3	4	5
(a) 講習会時間	短い⇔長い			4	2	1
(b) 講習会講義内容（プレゼン）	簡単⇔難		2	2	2	1
(c) 配布資料内容	簡単⇔難		2	2	2	1
(d) サンプルプログラム内容	簡単⇔難		2	1	3	1
(e) 満足度（平均 3.86）	不満⇔満足			2	4	1

講義内容の詳細については、ウェブページから資料及び録画ビデオをダウンロードできるのでそちらを参照いただきたい (<https://www.cc.u-tokyo.ac.jp/events/lectures/179/>)。また、従来実施していたデータ依存性を含むケースについては、3 日にわけてプログラムを再編成し、10 月以降に実施する予定である。

第 180 回お試しアカウント付き並列プログラミング講習会

「OpenFOAM 入門・キャビティ解析」実施報告

今野 雅

東京大学情報基盤センター客員研究員

2022 年 5 月 31 日 (火), PC クラスタコンソーシアム (実用アプリケーション部会・HPC オープンソースソフトウェア普及部会)、オープン CAE 学会との共催で、第 180 回お試しアカウント付き並列プログラミング講習会「OpenFOAM 入門・キャビティ解析」がオンラインで開催された。本講習会は、センターに設置されたスーパーコンピュータ (以降, スパコン) の利用促進とスパコンを用いた数値流体解析の普及を目的として実施されたものである。なお、本講習会はセンターのお試しアカウント付き並列プログラミング講習会として行われた 22 回目の OpenFOAM の講習会である。受講者は、大学・研究機関教職員 4 名、大学院学生 3 名、学部学生 1 名、企業の方 1 名であり、事前申込者 13 名、受講者合計は 9 名であった。センターが運営するスパコン Wisteria/BDEC-01 を用い、Wisteria/BDEC-01 の概要、利用方法、OpenFOAM の演習が 1 日終日の日程で行われた。当日のプログラムを表 1 に掲載する。なお、講習会終了後約 1 ヶ月有効なお試しアカウント (Wisteria-0, 最大ノード数 12, 最大実行時間 15 分) が受講者に与えられた。

表 1 講習会プログラム

【2022 年 5 月 31 日 (火)】	
10 : 00 - 12 : 00	概要説明
	イントロダクション (約 13 分間)
	Wisteria/BDEC-01 概要 (約 40 分間)
	OpenFOAM 概要 (約 45 分間)
13 : 00 - 18 : 00	キャビティ流れ演習
	講習会の準備 (約 7 分間)
	module による OpenFOAM の環境設定 (約 20 分間)
	解析対象 (約 8 分間)
	解析ケース (約 21 分間)
	blockMesh による格子生成 (約 49 分間)
	格子の可視化 (約 24 分間)
	解析条件の設定 (約 41 分間)
	ソルバ実行 (約 10 分間)
	解析結果の可視化 (約 10 分間)
	解析結果の検証 (約 12 分間)
	並列計算 (約 16 分間)
	演習課題 (約 7 分間)
	チュートリアルの実行 (約 6 分間)
	質疑応答

講習会終了後のアンケート集計結果(回答数 7)を表 2 に示すが、参加した満足度の平均は 5 点満点中、4.43 と高かった。また、参加者から表 3～5 に示すご意見を頂いた。今後の講習会の参考にしたい。

表 2 アンケート集計結果

評 点	講習会の時間		講習会の講義内容 (プレゼン)		配布資料の内容		サンプルプログラム 内容		参加した満足度	
1	短い	0	簡単	1	簡単	1	簡単	1	不満	0
2		0		1		0		1		0
3	適切	4	適切	4	適切	5	適切	4	普通	2
4		2		1		1		0		0
5	長い	1	難	0	難	0	難	1	満足	5
	平均	3.57	平均	2.71	平均	2.86	平均	2.86	平均	4.43

表 3 本講習会に対するご意見(原文ママ)

- 講師の方の質問への回答やトラブル対応など、大変素晴らしい講師の方でした。
- 次回までに勉強いたしますので、今後とも宜しくお願い致します。
- 午前中の概要説明は配布資料として参加者各自に参加するようにした方が良いです。午後の内容「キャビティ流れ演習」を朝から実施した方が良いと思います。それ以外演習の内容が適切と思います。
- 特になし

表 4 Zoom によるオンライン講習会で良かったこと(原文ママ)

- 自分のペースで学習できる
- 質問にすぐに回答していただける。個別トラブルを解消してくださる。
- 出張なしで受けられたこと
- 時間や場所などの制約がなく、講習会に参加しやすいこと
- 画面上で資料と操作が見やすく、理解しやすい。

表 5 Zoom によるオンライン講習会で悪かったこと(原文ママ)

- 特にございません。
- 途中で分かりにくいところ質問するのが難しいこと
- 特になし
- 特になし

第 11 回 JCAHPC セミナー（OFP 運用終了記念シンポジウム）

「ありがとう OFP:京から富岳への狭間で咲いた大輪の花」（ハイブリッド開催）

情報システム部情報基盤課 前田光教
情報基盤センター 住元真司

1 概要

2022 年 5 月 27 日（金） 第 11 回 JCAHPC セミナー（OFP 運用終了記念シンポジウム）「ありがとう OFP:京から富岳への狭間で咲いた大輪の花」¹を開催しましたので開催概要を報告します。

最先端共同 HPC 基盤施設（JCAHPC: Joint Center for Advanced High Performance Computing）は、複数の大学が共同してより大規模な HPC 基盤を構築するために国内で最初に筑波大学計算科学研究センターと東京大学情報基盤センターが共同で 2013 年に設立された組織です。JCAHPC では国内最大級の計算性能を有する Oakforest-PACS システム（OFP）を設計、導入し、2016 年 10 月より共同で運用を開始して以来、最先端の計算科学を推進し、我が国と世界の学術及び科学技術の振興に寄与してまいりました。JCAHPC は HPCI システム構成機関として「新型コロナウイルス感染症対応 HPCI 臨時公募課題」に計算資源を提供し、新型コロナウイルス感染症に関する研究推進に貢献しました。OFP は運用開始直後の 2016 年 11 月の TOP500 で「京」を上回り、国内最高性能を達成し、更に 2019 年 8 月末の「京」運用停止後、2021 年 3 月に「富岳」が正式に稼働を開始するまでの約 1 年半の間、実質的に National Flagship System としての役割を果たし、2022 年 3 月 31 日を以て運用を終了しました。

本セミナーは、JCAHPC を構成する筑波大・東大センター教員とその OB の他、OFP と同じ Intel Xeon Phi 搭載システムを運用する北大・京大センター教員の皆様、OFP のヘビーユーザーの皆様からの講演者を迎え、JCAHPC 設立及びその準備段階からの 10 年あまりを振り返るとともに、将来へ向けた展望も紹介され、次世代機への取り組みも紹介されました。

開催は、Covid-19 の感染拡大によるオンライン開催継続後初めてハイブリッド形式で開催され、会場と遠隔参加による講演者と聴講者（申し込み 212 名、実参加者数、現地 23 名、遠隔 149 名）により有益な講演と議論が行われました。本報告では本セミナーの概要とハイブリッド開催概要を報告します。

資料と公開可能な講演動画については JCAHPCA セミナーWeb ページ(URL 脚注 1)より参照いただけますので是非ご覧ください。

2 第 11 回 JCAHPC セミナー概要報告

第 11 回 JCAHPC セミナーの全体構成は、来賓挨拶のあと、OFP の軌跡を示すため JCAHPC 設立から OFP 概要、成果、今後の展開を含めた 4 部構成となっています。以降、各部の講演について報告します。

¹ <https://www.cc.u-tokyo.ac.jp/events/jcahpc/11.php>

2.1 第1部 概要

第1部は田浦健次朗 東京大学情報基盤センター長による開会挨拶のあと、7名の来賓からビデオレターによる OFP 運用終了への祝辞と OFP システムの果たした偉大な功績について語って頂きました。

表 1 第1部プログラム

【第1部】	開会
座長	田浦健次朗（東京大学情報基盤センター/JCAHPC）
13:00 - 13:05	開会
13:05 - 13:25	来賓等挨拶（ビデオレター） ・永田恭介（筑波大学長） ・藤井輝夫（東京大学総長） ・池田貴城様（文部科学省研究振興局長） ・時田隆仁様（富士通株式会社代表取締役社長 CEO, CDXO） ・Mr. Joseph (Joe) Curley (Vice President & General Manager, Software Products & Ecosystem in Intel's Software & Advanced Technology Group, Intel Corporation) ・Dr. Soonwook Hwang (Principal Researcher, Korean Institute of Science & Technology Information (KISTI)/Vice President, Korean Society for Computational Science and Engineering (KSCSE)) ・Dr. Richard Gerber (Senior Science Advisor & High Performance Department Head, National Energy Research Scientific Computing Center (NERSC))

2.2 第2部 概要

第2部では、「OFP 概要」というテーマで、佐藤三久 初代最先端共同 HPC 基盤施設・施設長（現理化学研究所計算科学研究センター 副センター長）から JCAHPC 設立について2大学での2011年からの設立までの取り組みについて苦労話を含めて紹介いただきました。JCAHPC 設立の理念として「京と対抗しつつ、補完するスパコンの拠点」を目指し、設立・運用を通して理念を具現化したことを紹介いただきました。

続く、朴泰祐 現 JCAHPC・副施設長／筑波大学計算科学研究センター センター長からは、OFPの概要と導入について、T2K から JCAHPC への繋がったこと、その生い立ちからテクノロジー選定の苦労、システム名の由来、ハードウェア、ソフトウェア、採用したテクノロジーの効果を含めて幅広く紹介されました。アプリケーション開発だけでなく、システムソフトウェアを含めた成果を創出したこと、京コンピュータの運用終了後、日本国内で HPCI 第2階層の代表的システムとして HPCI を支えた点、複数大学の共同調達による国内最高システムの構築というパイロットシステムの意味をもつことを紹介いただきました。

続いて、深沢圭一郎 京都大学学術情報メディアセンター 准教授からは京都大学における OFP と同じプロセッサ Xeon Phi を持つ Camphor2 システムとそのシステム運用について紹介いただきました。Camphor2 では OFP とは異なるインターコネクトである Aries+Dragonfly が採用され、その性能最適化の苦労、他システムの性能比較について紹介いただきました。

最後は、岩下武史 北海道大学情報基盤センター 教授から、同じ Xeon Phi を搭載した Polaire システムの設計思想と活用について紹介いただきました。高い省電力性能に加え、オープンスーパーコンピュータ (T2K) の思想を引き継ぐと共に従来スーパーコンピュータのソフトウェア資産を継承する思想を元に Xeon Gold と Xeon Phi を搭載する2種のシステム構成としたこと、Polaire

システム上での活用研究について紹介いただきました。

続くディスカッションでは Xeon Phi 搭載 3 システム間で相互の運用法ほか、有意義で幅広い議論が行われました。相互のセンター間での継続した技術交流があまり行われなかったとのことで、今後のセンター間でのアプリケーション開発と運用を含めた技術交流を進めることによるシステム改善が重要であるとの意見が出されました。

表 2 第 2 部プログラム

【第 2 部】	OFP 概要	参加
座長	建部修見（筑波大学計算科学研究センター/JCAHPC）	現地
13:25 - 13:45	佐藤三久（理化学研究所計算科学研究センター） JCAHPC 設立と理念	現地
13:45 - 14:05	朴泰祐（筑波大学計算科学研究センター/JCAHPC） OFP の概要と導入について	現地
14:05 - 14:20	深沢圭一郎（京都大学学術情報メディアセンター） Camphor2 : OFP と同じ Xeon Phi KNL を搭載し、OFP より少し長く運用される 京都大学スパコン	遠隔
14:20 - 14:35	岩下武史（北海道大学情報基盤センター） 北大スパコン Polaire ー設計思想と活用研究ー	遠隔
14:35 - 14:50	建部修見（筑波大学計算科学研究センター/JCAHPC） （ディスカッション）	ハイブリッド
14:50 - 15:00	休憩	

2.3 第 3 部 概要

第 3 部では、「OFP による成果」というテーマで、中島研吾 東京大学情報基盤センター 教授 /JCAHPC 研究開発部門長から OFP における研究の支援活動と成果について、HPC チャレンジ、HPCI における Covid-19 対応臨時公募課題、システム運用施策、利用統計を踏まえて紹介されました。利用状況については、運用当初は伝統的な科学技術演算主体であったものが 2019 年度からデータ科学・データ同化分野での利用が増え始めるなど様々な分野で活用されている一方、工学・ものづくり関連の計算は全般的に利用が少ないといった課題提示がありました。講演では OFP 上で行われたアプリケーションとシステムソフトウェアにおける様々な研究成果について紹介され、数多くの成果が創出されたことが示されました。全体的にみて、JCAHPC 自体としての研究活動はやや不活発であったとのことで、OFP-II に向けては 2 件 JHPCN 共同研究課題を新規提案、採択されたとのことで、次世代機に向け JCAHPC 自体の研究活動を活性化すると報告でした。

2 件目の成果として、理化学研究所計算科学研究センターの三好建正 先生によるリアルタイムゲリラ豪雨予測実験について OFP 上での研究成果を紹介いただきました。講演では 2019 年の京コンピュータ共用終了後、OFP 上でのゲリラ豪雨予測システム開発過程と 2020 年に行われたリアルタイム実証実験の内容をシステムのワークフローからジョブスケジューリングを含めた計画を紹介いただきました。2021 年の富岳の運用開始後は富岳上でのリアルタイム実証実験を開始されました。国内 Flagship システムが存在しない状況下における HPCI 第 2 階層のシステムである OFP が研究の継続性確保に大変重要な役割を果たしたと紹介いただきました。

3 件目の成果として、日本原子力研究開発機構の井戸村泰宏 先生より、富岳への適用を想定

した OFP 上でのアプリケーション最適化手法について、特にメニーコア向けの省通信型行列解法技術を紹介いただきました。OFP と富岳はシステムアーキテクチャ的に多くの共通点があり、OFP で事前に OFP で検証・最適化することにより、早期の最適化課題の認識と、富岳運用開始時に早い成果創出が可能になったとのことです。実際のアプリケーション評価においては、HPC チャレンジを活用頂き OFP 全系での強スケーリング評価の結果、8000 ノード規模での良好な性能向上を達成することができました。本成果は富岳に移植され高性能計算が実現できたとのことで OFP が有効に活用された事例となります。

表 3 第 3 部プログラム

【第 3 部】	OFPIによる成果	参加
座長	高橋大介（筑波大学計算科学研究センター/JCAHPC）	現地
15:00 - 15:20	中島研吾（東京大学情報基盤センター/JCAHPC） Oakforest-PACSによる研究成果の概要	現地
15:20 - 15:40	三好建正（理化学研究所計算科学研究センター） 世界初リアルタイム30秒更新ゲリラ豪雨予測実験	遠隔
15:40 - 16:00	井戸村泰宏（日本原子力研究開発機構(JAEA)） Oakforest-PACSにおける大規模CFD解析向け省通信型行列解法の開発	遠隔
16:00 - 16:10	休憩	

2.4 第 4 部 概要

第 4 部では、「次の一手」というテーマで、次世代計算基盤に関する計画を筑波大学、東京大学、理化学研究所、JCAHPC の各センター代表者から語っていただきました。

- 筑波大学計算科学研究センター（朴教授）：引き続き JCAHPC のシステムでは幅広いユーザ層を支えるシステムを導入し、筑波大学ではやや尖ったユーザも視野に入れ高性能・省電力な先進的な加速器型スーパーコンピュータを導入していく。次期システムにおいては、加速器型プロセッサの高い計算能力がメモリ量不足のために活かしきれない点に着目し、低コストで大容量・高性能・低消費電力である Persistent Memory (PMEM) を導入することで大容量データ処理に向けたシステムを導入する。システムに合わせてストレージ周りのシステムソフトウェアも整理する。次期導入予定の Cygnus-BD と呼ばれるシステムは次世代 Xeon+PMEM+NVIDIA H100 から構成される 120 ノードシステムで 2022 年 11 月に稼働する予定。
- 東京大学情報基盤センター（中島教授）：2021 年に運用開始した Wisteria/BDEC-01 システムの設計理念である Society5.0 を具現化する計算・データ・学習 (S+D+L) 融合のためのプラットフォーム開発とソフトウェア基盤である h3-Open-BDEC を開発整備していく。様々な領域の問題に適用可能にするため更なる異種システム間連携を実現・発展させていく。複数システム間連携のデータ基盤として大規模共通ストレージ Ipomoea と MDX を活用する。2027/10-2028/4 に運用開始予定の Wisteria/BDEC-02 に向けては AI for HPC を更に発展させ DPU、IPU、Quantum-Inspired Device を取り入れた異種システム間連携も視野にいれ検討していく。高性能・省電力・脱炭素化を推進するには演算加速器搭載は不可避

であるためユーザ利用の多様化に対応可能なプログラム開発環境、移植が課題である。今後も、HPCI、JHPCN 構成機関とも協力して「計算・データ・学習」融合推進を継続すべく研究活動を実施していく。

- NGACI (近藤教授)： 2028 年頃に利用可能な技術を用いた次世代情報基盤 (富岳 NEXT) の検討を進めている。2020 年 11 月に White Paper 1.0 版を発行し、現在 1.1 版を準備中である。プロセッサについてはこれまでの汎用プロセッサのみでは性能とシステム電力要求を満たせないため、汎用と「広義」のアクセラレータを前提にアーキテクチャ、システムソフトウェア、アプリケーション・ライブラリ・アルゴリズム間のコデザインを進めている。これまでに、汎用システム型とアクセラレータ主体・ヘテロジニアス型他のアーキテクチャを想定した検討を進めている。今年度、「次世代計算基盤に関する FS プロジェクト」も公募開始となっており、具体的なシステム検討も加速される。富岳 NEXT だけでなく、第二階層のシステムを含めた次世代計算基盤の議論が重要になる。今後もアプリケーション・システムのコデザインも相互に協力・連携して次世代計算基盤を実現していきたいので一層の協力をお願いしたい。
- JCAHPC (塙教授)： OFP の次世代システムとして、システム名称を Oakforest-PACS II (以下、OFP-II) と決定し、2024 年 4 月稼働を目指して設計を進めている。現状、システム規模は 200+PFLOPS をターゲットに汎用メニーコア CPU+一部 GPU システムとして、HPCI 第二階層システムをリードする Upscale Commodity Cluster Machine(UCC)、かつ、Technology Path-Forward Machine(TPF)に位置づけられるシステムとする。システム設計では OFP 等、従来のシステムユーザの継承性を維持しつつ、新しい応用に向けて計算+データ+学習融合によるシミュレーションの高度化を進めていく。今回、ユーザ需要と電力対性能の観点から OFP-II では初めて GPU を導入するため、OFP-II プロトタイプシステムとして Wisteria-Mercury システムを導入し、GPU 選定とその GPU 向けのアプリケーション準備を加速する。加えて OFP-II に向けて JCAHPC 内での協調研究として JHPCN (学際大規模情報基盤共同利用・共同研究拠点) 課題に今年度 2 件採択された。これらの成果を OFP-II 設計へのフィードバックと調達時のベンチマーク作成に活用する。これらの取り組みにより、広範囲なシステムユーザに対して徐々に GPU への移行を進め、次世代機以降への新たなシステム設計を進めていく。

表 4 第 4 部プログラム

【第 4 部】	次の一手	参加
座長	中村宏（東京大学情報基盤センター/JCAHPC）	遠隔
16:10 - 16:30	朴泰祐（筑波大学計算科学研究センター/JCAHPC） 筑波大学におけるビッグメモリスーパーコンピュータの導入	現地
16:30 - 16:50	中島研吾（東京大学情報基盤センター/JCAHPC） 東京大学情報基盤センターの現状と今後の計画	現地
16:50 - 17:10	近藤正章（慶應義塾大学/理化学研究所計算科学研究センター） 次世代先端的计算基盤の開発に向けたNGACIでの取り組み	遠隔
17:010 - 17:30	塙敏博（東京大学情報基盤センター/JCAHPC） Oakforest-PACS IIに向けて	現地
17:30 - 17:45	中村宏（東京大学情報基盤センター/JCAHPC） （ディスカッション）	ハイブリッド
17:45 - 17:50	石川裕（国立情報学研究所(NII)） OFP運用終了に当たって	現地
17:50 -	中村宏（東京大学情報基盤センター/JCAHPC） 閉会あいさつ	遠隔
	セミナー終了	

2.5 OFP 運用終了に当たって

最後に、JCAHPC 創立当時の東京大学情報基盤センター センター長であられた石川裕 教授(現、国立情報学研究所 教授)から、OFP 運用終了に当たっての挨拶がありました。

OFP の果たした役割については、一連の講演で十分語られている一方、当初目指した計算科学、数理科学研究者と共に数値計算ライブラリ、システムソフトウェア、および、アプリケーションをコデザインすると共に人材育成機関としての JCAHPC を見た場合に、システムソフトウェア関連では OFP で McKernel 開発に利用され、一部利用された点は良かったが、運用で十分活用されるまでに至らなかったこと。2 大学が行った共同調達・運用・研究の取り組みが他の大学・研究機関に広まらなかったことが残念であり、教育と若手の育成を含め今後の取り組みに期待したいとの言葉を頂きました。

最後に、好きな言葉として日本酒「獺祭 磨 その先へ」の Web 紹介の文章から「いかに困難が予想されても、いかに現在が心地好くても、その先へ、我に安住の地はなし。」を引用され挨拶を締めくくられました。

3 第 11 回 JCAHPC セミナーのハイブリッド運営について

2020 年度からの第 9 回及び第 10 回 JCAHPC セミナーでは、新型コロナウイルスの影響によりオンラインによる開催を余儀なくされました。2022 年度はその影響は小さくなったため、感染拡大防止対策を考慮し、講演者の多くが会場から参加し、視聴者はオンラインから参加するハイブリッド形式による開催を実施しました。今回のセミナー開催では講演者、座長、聴講者を含めて現地参加、遠隔参加の形式でした。(講演者、座長については表 1、表 2、表 3、表 4 参照)

本節では、ハイブリッド形式による開催を実現するためのウェブ会議システムと会場講義システムとを組み合わせる配信した手法を紹介します。

3.1 ハイブリッド配信

ウェブ会議システムは Zoom² ウェビナーを利用しました。

Zoom ウェビナーはホストの他、パネリストを中心とするセミナー形式のオンラインによる画面共有と音声配信を備えているため、講演者及び視聴者がそれぞれの PC 等から簡単に参加することができます。シンポジウムのハイブリッド配信では、その画面と音声を会場の講義システムに入出力することで、会場でのプロジェクタ投影、スピーカー、マイクと連携したいと考えました。その連携のためには Zoom 配信用の PC から画面や音声を入出力できる仕組みが必要となります。しかしながら、本セミナーで利用した会場は、新型コロナウイルスの影響によりウェブ会議システムが隆盛した以前に設計された講義システムであり、残念ながら PC からの入出力インターフェースがありませんでした。ただし、設計当時に多く利用されていた Polycom³ のテレビ会議システムが導入されており、Zoom とのオンライン接続を Polycom システムの H. 323 機能を利用し連携させることにしました。

また、講演者の資料は会場やオンラインに関わらず講演者本人の PC から Zoom の画面共有機能を利用することとし、会場講義システムのプロジェクタ投影専用 PC (Zoom には視聴者として参加) を通すことで講義システムの切り替えをなくし、進行の妨げとならない工夫をしました。会場の講演者は壇上にいながら Zoom を共有することにはなりますが、オンラインの講演者と操作を同じくできる利便もありました。また、来賓等挨拶はビデオレターとし録画したビデオファイルをまとめて Zoom の画面共有で配信することとしました。

3.2 Zoom と会場講義システムとの連携

東京大学がライセンス契約する Zoom は 300 名までのミーティングが可能であり、1000 名までのミーティングまたは 3000 名までのウェビナーが申請可能となっていますが、個別に有料オプション購入はできないため、Polycom との連携に必須の H. 323 オプションが表示されませんでした。そのため、スーパーコンピューティング部門が契約する Zoom で 500 名までのウェビナーと H. 323 オプションを利用することとしました。

いずれのオプションも月単位で契約/解除が可能であるため、必要な時に必要な分を調達できる柔軟性があることも確認でき、オプション契約した時点から利用可能となるため、早速、会場講義システムとのテストを実施しました。

Zoom には予めテスト用ウェビナーをスケジュールし、会場 Polycom からアニュアルに従い Zoom H. 323 連携用の地域 IP アドレス (Japan あり) に接続し、ミーティング ID を入力します。「IP アドレス##ミーティング ID」のように 1 度の入力も可能です。入力の注意点として、Polycom のリモコン数字キーを利用する場合はタッチトーン (DTMF) と呼ばれるボタンを画面から選択してからとなります。パスワードを設定している場合はパスワードの入力も求められます。また、通常 Polycom はグローバル IP アドレスによる運用をしていると思われますが、NAT やプロキシなどを経由する場合は注意が必要になります。

² <https://zoom.us/>

³ <https://www.poly.com/>

ありがとうございますOFF ハイブリット配信（案）

会場設備 (プロジェクタ、カメラ、ディスプレイ)

会場Polycom

連携

配信Zoom

配信

Zoomウェビナー

遠隔講演者

H.323/SIP 中継サーバ

配信 (緊急時)

会場Zoom (バックアップ配信用)

Zoomのオプション契約

- h.323/sipプラグイン
- ウェビナー

※Proだと100人が上限のため

3.3 会場環境の整備

本セミナー会場の講義システムは、PC を接続のために HDMI と RGB(D-SUB)が準備されています。いずれも会場プロジェクタ利用には問題ありませんでしたが、画面をPolycomに通すとRGBしか連携できず、解像度の調整などしましたが、HDMI 入力に対応しないとの結論にいたりしました。

マイクは会場からと Zoom から発話を想定した音量を調整し、ハウリングしないか、ハウリングが起こる環境をチェックしました。

- 照明

ハイブリッドに限りませんが、講演者の紹介は明るく、講演中はプロジェクタ付近を暗くするなど調整するため、照明スイッチを確認し、当日の担当者に切り替えのタイミングを指示する必要があります。

- 会場カメラによる画像配信

講演画面だけではなく、会場の様子を伝えるために、講義システムの持つカメラ映像を配信することにしました。

この他チェックすべき点については、この後に記載する「要確認チェックリスト」にまとめています。

3.4 開催当日

テストでは想定できず、当日開催してから課題となった点があります。

開催当日は2時間前に会場入りしテストした環境を再現します。今回の再現では、テストした環境を記憶だけに頼ったため一部再現できず課題が残ることになりました。このため、「要確認チェックリスト」を作成する対策をしました。

講演者がマイクをオフのまま始めてしまい、Zoom 視聴者には全く聞こえない時間がありました。また、講演者の声や音質、マイクの使い方にバラツキがあり、会場は生の声が届くため気にならない場合もありますが、Zoom 側は聞き取りづらいとの報告がありました。これは、録画データでも同じ傾向になり、セミナー後の録画公開のための編集で調整を必要とします。急遽ピンマイクを用意し音量の問題を小さくすることができましたが、テストではピンマイクも確認すべきでした。

講演資料の画面共有では、会場プロジェクタに投影される画面の左右が切れる問題が発生しました。これは一部テスト環境を再現できなかった課題であり、セミナー開催中に解決できませんでした。会場講義システムのスイッチャは様々なデバイスをプロジェクタに投影できますが、PCではなく Polycom が選択されていたことが原因でした。これに連動し、会場カメラとの併用が可能だったにも拘わらず、一部のタイミングで講演者の様子を配信できませんでした。Zoom ウェビナーには資料の画面共有の間、講演者が PC のカメラをオンにすることで視聴者の画面上に小さく講演者カメラがオプトインされる仕組みを活用できませんでした。

テスト環境では分からなかった点に、Zoom 視聴者にはハウリングしたような響く音声になり、これも開催中に解決できませんでした。セミナー後に、会場の天井に集音マイクがあることが原因でした。

3.5 要確認チェックリスト

テストと当日との接続構成を同じくできるよう、本セミナーで確認した課題を考慮したチェックリストを作成しました。皆様の環境により細かには違いはあるでしょうが、ウェブ会議システムと会場講義システムとの連携で必要な項目を列記しましたので、お役に立ていただければ幸いです。

表 5 要確認チェックリスト

作業項目リスト(会場設備事前確認)	備考	有無
会場講義システムと配信用PC(zoomなど)との接続環境		
会場カメラ入出力(RGB(D-SUB)、HDMI、USB、ピン、パソコン)	講義システムに直接接続できるか	
会場マイク入出力(ミニジャック(2極、3極)、USB、ピン)	講義システムに直接接続できるか	
ハイブリッド配信の環境が複数ある(バックアップ含む)か検討		
配線の取り回しによる講演者や補助者の位置確認	電源、ケーブル	
会場照明環境のスイッチング(全灯、プレゼン向けなど)	プリセット式か	
カメラワークや複数の場合の切替	リモコン	
マイクの音量、本数や配置	集音マイクの確認	
ハイブリッド配信でのハウリング条件確認	キャンセラー	
会場プロジェクタへの投影画面(解像度、ケラレの確認)		
ハイブリッド配信への画面共有(解像度、ケラレの確認)	動画は音声も同時に送れること	
座長席、統括担当席、補助者席、受付場所	人員確保	
担当者リストの作成		
感染拡大防止対策(注意事項、消毒、配置、換気など)		

3.6 ハイブリッド形式開催のまとめ

今回、新型コロナウイルスによる実会場でのセミナー開催中断後、初めてのハイブリッド形式によるセミナー開催を試みました。初めての試行でわからないことも多く不具合もありましたが、セミナー開催を実施することができました。これも参加者・関係者のみなさまのご協力の賜物と考えております。ここに感謝させていただきたいと思います。

今回、ハイブリッド開催における運用の知見については開催要領、要確認チェックリストを作成し、次回以降の開催毎にブラッシュアップしていき、開催ノウハウを逐一蓄積していく予定です。参考にされたい方がいらっしゃいましたら、お気軽にお問い合わせくださるようお願いいたします。

4 おわりに

今回、第 11 回 JCAHPC セミナーを企画・開催し、JCAHPC 設立経緯から、運用、次世代計画まで、テーマが広範囲だけでなく、設立理念からシステム調達、運用、大規模アプリケーション最適化技術まで内容も深く紹介され、ハイブリッド形式により活発な議論がありました。初めてのハイブリッド開催ということで不具合もありましたが、講演頂いた先生方、多くの参加された皆様のおかげで素晴らしいシンポジウムとなりました。ここに感謝する次第です。

本開催は、0FP の運用終了への感謝の記念シンポジウムでしたが、シンポジウムを通して、これは新たなチャレンジへの始まりであることを再認識しました。今後も、先人の設立理念を継承・発展させ、現状に満足せず、我々のミッションである革新的なシステムとサービス提供し、その活動を広く国内外に広げるために尽力していく次第ですので、みなさまのご協力よろしくお願いいたします。

原稿募集

本誌では利用者の皆様からの原稿を募集しています。以下の執筆要項に基づいて投稿してください。

執筆要項

- 1 内容は、本センターのスーパーコンピュータシステムの利用者にとって有意義な情報の提供となる原稿とします。
- 2 掲載可否については当編集委員会で決定させていただきます。
- 3 掲載可とした投稿原稿に対して、加除訂正を行うことがあります。
- 4 原稿枚数には特に指定はありませんが、シリーズに分割することもあります。
- 5 プログラムの実例が大量になる場合（概ね 1 頁を超える）は、本文には一部のみを記述し、投稿者の Web ページ等に全体を掲載し、その URL を引用するようにしてください。
- 6 原稿は横書きにしてください。
- 7 原稿は、A4 サイズで、ページの余白は上下 20mm、左右 26mm、ヘッダー15mm、フッター10mm に設定してください。詳しくは原稿様式をご参照ください。PDF 形式（フォント埋め込み）の完全原稿を電子メールにて uketsuke@cc.u-tokyo.ac.jp までご提出願います。
- 8 採用された原稿は、本センターの Web ページに掲載いたします。
<https://www.cc.u-tokyo.ac.jp/public/news.php>

【スーパーコンピュータシステム・ストレージシステム利用案内】

お知らせ	Web ページ
サービス案内、運転状況など	https://www.cc.u-tokyo.ac.jp/
公開鍵登録、マニュアル閲覧など	https://wisteria-www.cc.u-tokyo.ac.jp/ (Wisteria/BDEC-01) https://obcx-www.cc.u-tokyo.ac.jp/ (Oakbridge-CX) https://ipomoea-www.cc.u-tokyo.ac.jp/ (Ipomoea-01)

お問い合わせ内容	お問い合わせ先
利用申込関係	スーパーコンピュータシステム・ストレージシステム 利用申込書提出先 uketsuke@cc.u-tokyo.ac.jp 東京大学情報システム部 情報戦略課研究支援チーム
プログラム相談・システム利用に関する質問	https://www.cc.u-tokyo.ac.jp/supports/contact/#SOUNDAN
システムに関する要望・提案	voice@cc.u-tokyo.ac.jp

【IP ネットワーク経由時のホスト名】

シ ス テ ム	ホ ス ト 名
Wisteria/BDEC-01 スーパーコンピュータシステム (Odyssey/Aquarius)	wisteria.cc.u-tokyo.ac.jp 以下のホストの何れかに接続します※ wisteria0{1-6}.cc.u-tokyo.ac.jp
Oakbridge-CX スーパーコンピュータシステム	obcx.cc.u-tokyo.ac.jp 以下のホストの何れかに接続します※ obcx0{1-6}.cc.u-tokyo.ac.jp
Ipomoea-01 システム (大規模共通ストレージシステム第1世代)	ipomoea01.cc.u-tokyo.ac.jp 以下のホストの何れかに接続します※ i01login{1-2}.cc.u-tokyo.ac.jp

※どのホストに接続しても同じです。

【編集】

東京大学情報基盤センタースーパーコンピューティング研究部門
 東京大学情報システム部情報基盤課スーパーコンピューティングチーム
 〃 情報戦略課研究支援チーム

【発行】

東京大学情報基盤センター
 〒277-0882 千葉県柏市柏の葉6-2-3
 (電話) 04-7133-4663 (ダイヤルイン)

目 次

センターから

サービス休止等のお知らせ	1
システム変更等のお知らせ	3
大規模共通ストレージシステム (第1世代)	
運用に関するお知らせ(再掲)	6
Oakbridge-CXとWisteria/BDEC-01で稼働する技術計算言語	
MATLAB のご紹介	9
「MATLAB」提供開始:「計算・データ・学習」融合, AI-for-HPC/AI-for-Scienceの実現へ向けて	13
スーパーコンピュータシステム「大規模HPCチャレンジ」	
課題募集のお知らせ	17
研究成果登録のお願い	20
4月・5月のジョブ統計	21

研究報告

Wisteria/BDEC-01利用事例 (6) マルチプログラム連成ライブラリ h3-Open-UTIL/MP(2/2)	24
--	----

ユーザーから

ミニマルスパン・チャネル乱流の直接数値計算による乱流伝熱解析	37
Geomechanical modeling for subsurface CO2 storage with active pressure management : A preliminary study	40
Efficient Simulation of Ferromagnetic Spin Configuration	44

教育活動報告

第176回お試しアカウント付き並列プログラミング講習会 「MPI基礎:並列プログラミング入門」実施報告	46
第177回お試しアカウント付き並列プログラミング講習会 「MATLABの実行方法」実施報告	50
第178回お試しアカウント付き並列プログラミング講習会 「GPUプログラミング入門」	52
第179回お試しアカウント付き並列プログラミング講習会 「OpenMPによるマルチコア・メニョコア並列プログラミング入門」 (Wisteria/BDEC-01 (Odyssey, A64FX搭載)) (オンライン)	53
第180回お試しアカウント付き並列プログラミング講習会 「OpenFOAM入門・キャビティ解析」実施報告	55
第11回JCAHPCセミナー (OFF運用終了記念シンポジウム) 「ありがとうOFF:京から富岳への狭間で咲いた大輪の花」 (ハイブリッド開催)	57

原稿募集	67
------	----