（オンライン） ISSN 2436—4940

スーパーコンピューティングニュース

Vol.24 No.1, 2022.1

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）

コース	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	【Wisteria-O/A】 申込 1 セット当り 60,000 円 （8,640 トークン）		グループ 1 セット当り /work 2TB 利用者当り /home 50GB	【Wisteria-O】 最大ノード数 2,304 ノード 【Wisteria-A】 最大 GPU 数 64
公募制度による申込 （Wisteria-A における「GPU 専有申込」,「ノード固定」も可能）	【Wisteria-O】 申込 1 セット当り 60,000 円 （8,640 トークン）	【Wisteria-O】 申込 1 セット当り 72,000 円 （8,640 トークン）	グループ 1 セット当り /work 2TB 利用者当り /home 50GB	
	【Wisteria-A】 申込 1 セット当り 180,000 円 （25,920 トークン）	【Wisteria-A】 申込 1 セット当り 216,000 円 （25,920 トークン）	グループ 1 セット当り /work 6TB 利用者当り /home 50GB	
GPU 専有申込 （Wisteria-A のみ，公募制度による申込可能）	【Wisteria-A】 申込 1 セット当り 270,000 円 （25,920 トークン）	【Wisteria-A】 申込 1 セット当り 324,000 円 （25,920 トークン）	グループ 1 セット当り /work 6TB 利用者当り /home 50GB	1, 2, 4GPU のみ申込可，申込単位は下表参照，利用期間は 1 ヶ月単位で設定可
ノード固定 （Wisteria-A のみ，公募制度による申込可能）	【Wisteria-A】 申込 1 セット当り 2,160,000 円 （207,360 トークン）	【Wisteria-A】 申込 1 セット当り 2,592,000 円 （207,360 トークン）	グループ 1 セット当り /work 48TB 利用者当り /home 50GB	1 ノード 8GPU 1 年分，1 セットのみ申込可，利用期間は 1 ヶ月単位で設定可
トークン追加	5,000 円 （720 トークン）	6,000 円 （720 トークン）		
ディスク追加	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。

※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ノード時間積の換算表」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(2020 年 4 月 1 日)

コース		負担金額(税込)		ディスク容量	備考
		大学・公共機関等	企業		
パーソナルコース		申込 1 セット当り, 最大 3 セットまで 100,000 円 (8,640 ノード時間)		申込 1 セット当り /work 4TB 利用者当り /home 50GB	最大ノード数 256 ノード
グループコース	一般申込	申込 1 セット当り 100,000 円 (8,640 ノード時間)	申込 1 セット当り 120,000 円 (8,640 ノード時間)	グループ 1 セット当り /work 4TB 利用者当り /home 50GB	最大ノード数 256 ノード
	ノード固定	申込 1 セット当り 150,000 円 (8,640 ノード時間)	申込 1 セット当り 180,000 円 (8,640 ノード時間)		
トークン追加		8,400 円 (720 ノード時間)	10,000 円 (720 ノード時間)		
ディスク追加		6,480 円/(1TB*年)			1TB 単位で申込可 (/work のみ)

※トークン消費係数は 1.00 である。
※括弧内のノード時間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ノード時間積の換算表」を参照。
※ノード固定の申し込みには審査を要する。
※/home のディスク容量はパーソナルコースやグループコースに複数所属していても利用者当り 50GB 固定。

Oakforest-PACS スーパーコンピュータシステム 利用負担金表(2020 年 4 月 1 日)

コース	負担金額(税込)		ディスク容量	備考
	大学・公共機関等	企業		
パーソナルコース	申込 1 セット当り, 最大 6 セットまで 50,000 円 (8,640 ノード時間)		申込 1 セット当り /work 1TB 利用者当り /home 50GB	最大ノード数 2,048 ノード
グループコース	申込 1 セット当り 50,000 円 (8,640 ノード時間)	申込 1 セット当り 60,000 円 (8,640 ノード時間)	グループ 1 セット当り /work 1TB 利用者当り /home 50GB	最大ノード数 2,048 ノード
トークン追加	4,200 円 (720 ノード時間)	5,000 円 (720 ノード時間)		
ディスク追加	6,480 円/(1TB*年)			1TB 単位で申込可 (/work のみ)

※トークン消費係数は 1.00 である。
※括弧内のノード時間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ノード時間積の換算表」を参照。
※/home のディスク容量はパーソナルコースやグループコースに複数所属していても利用者当り 50GB 固定。

トークン移行におけるノード時間積の換算表

移行元 ＼ 移行先	Oakforest-PACS システム	Oakbridge-CX システム	Wisteria/BDEC-01 システム
Oakforest-PACS システム	ー	0.5	0.8
Oakbridge-CX システム	2	ー	1.6
Wisteria/BDEC-01 システム	1.2	0.6	ー

移行先に追加されるトークン量（ノード時間）＝移行トークン量×係数

注意事項（Wisteria/BDEC-01, Oakbridge-CX, Oakforest-PACS 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。
- パーソナルコース（ただし，本センターのスーパーコンピュータシステムに初めて登録された利用者）においてのみ，利用開始月の翌月末日までに利用を中止することができる。利用負担金はパーソナルコースの利用期間1ヶ月の金額を適用し，請求する。
- 利用負担金は，原則として利用開始月に応じ，以下の月の初旬に一括して請求する。
  - 利用開始月が 4 月から 7 月までは 10 月，8 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，7 月初旬の請求となる。
- 利用負担金額が減額となる変更はできない。
- コース間の変更については，利用負担金が増額になる場合のみ別途相談に応じる。（ただし，利用者番号変更の場合がある。）
- グループコースのディスク量は，グループ全体の上限値である。

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) 正式運用期間	制限時間 (経過時間) 試験運用期間	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	30 分	28	○	○
short-o	1 ～ 72 (3,456)	8 時間	4 時間	28	○	○
(regular-o)						
small-o	1 ～ 144 (6,912)	48 時間	12 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	6 時間	〃	○	○
priority-o	1 ～ 288 (13,824)	48 時間	12 時間	28	○	○
challenge-o	1 ～ 7,680 (368,640)	24 時間	-	28	★	★
(interactive-o) ※4						
interactive-o_n1	1 (48)	30 分	30 分	28	○	○
interactive-o_n12	2 ～ 12 (576)	10 分	10 分	〃	○	○
prepost	1 (56)	6 時間	3 時間	340	○	○
prepost1_n1 ～ prepost4_n1	1 (56)	1～6 時間	1～3 時間	340	○	○
prepost1_n4	1 ～ 4 (224)	1～6 時間	1～3 時間	340	○	○
prepost1_n8	1 ～ 8 (448)	1～6 時間	1～3 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) 正式運用期間	制限時間 (経過時間) 試験運用期間	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	30 分	448	○	○	○	○
short-a	1 ～ 2 ノード (16)	2 時間	1 時間	448	○	○	○	○
(regular-a)								
small-a	1 ～ 2 ノード (16)	48 時間	12 時間	448	○	○	○	○
medium-a	3 ～ 4 ノード (32)	〃	〃	〃	○	○	○	○
large-a	5 ～ 8 ノード (64)	24 時間	6 時間	〃	○	○	○	○
share-debug	1, 2, 4 GPU	30 分	30 分	56	○	○	○	○
share-short	1, 2, 4 GPU	2 時間	1 時間	56	○	○	○	○
(share)								
share-1	1 GPU	48 時間	12 時間	56	○	○	○	○
share-2	2 GPU	〃	〃	〃	○	○	○	○
share-4	4 GPU	24 時間	6 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	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	パーソナルコース	グループコース	
					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”

Oakforest-PACS スーパーコンピュータシステム ジョブクラス制限値(2018 年 4 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	パーソナルコース	グループコース
debug-cache	1 ～ 128 (8704)	30 分	82	○	○
debug-flat	1 ～ 128 (8704)	30 分	96	○	○
(regular-cache)					
small-cache	1 ～ 128 (8704)	48 時間	82	○	○
medium-cache	129 ～ 512 (34816)	〃	〃	○	○
large-cache	513 ～ 1024 (69632)	〃	〃	○	○
x-large-cache	1025 ～ 2048 (139264)	24 時間	〃	○	○
(regular-flat)					
small-flat	1 ～ 128 (8704)	48 時間	96	○	○
medium-flat	129 ～ 512 (34816)	〃	〃	○	○
large-flat	513 ～ 1024 (69632)	〃	〃	○	○
x-large-flat	1025 ～ 2048 (139264)	24 時間	〃	○	○
challenge	1 ～ 8208 (558144)	24 時間	82 / 96	★	★
(interactive-cache) ※4					
interactive_n1-cache	1 (68)	2 時間	82	○	○
interactive_n16-cache	2 ～ 16 (1088)	10 分	〃	○	○
(interactive-flat) ※4					
interactive_n1-flat	1 (68)	2 時間	96	○	○
interactive_n16-flat	2 ～ 16 (1088)	10 分	〃	○	○
prepost	1 (28)	6 時間	222	○	○

★ 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-cache/flat，debug-cache/flat を小文字で指定する
regular-cache/flat 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 インタラクティブジョブの起動は， pjsub --interact -g グループ名 -L“rscgrp=キュー名,node=ノード数”
（キュー名は interactive-cache/flat，トークン消費なし）

巻頭言：「計算・データ・学習」融合，２年目の始まり

中島研吾
東京大学情報基盤センター

新しい年，2022 年の初めにあたって，皆さまとご家族，ご友人，周囲の皆さまのご健康を心からお祈り申し上げます。新型コロナウイルスとの戦いは一進一退を繰り返しておりますが，東京大学情報基盤センター（当センター）の活動レベルも昨年 12 月から「レベル A」とし，感染拡大に最大限の配慮をしつつ，ほぼ通常に近い形での活動レベルに戻しております。当センター，スパコン関連各社との緊密な協力により，利用者の皆さまへの影響は最小限に留まっていると考えておりますが，細かいところでご不便をおかけしていること，改めてお詫び申し上げるとともに，引き続きのご理解，ご協力をよろしくお願いいたします。

本年度も昨年度に引き続いて HPCI[1]（革新的ハイパフォーマンス・コンピューティング・インフラ ）による臨時課題募集「新型コロナウイルス感染症対応 HPCI 臨時公募課題[2]」に協力しており，2021 年末現在で，全 4 課題のうち 3 課題が当センターのスパコンシステム（Oakbridge-CX：2 課題，Oakforest-PACS：1 課題）を使用しています。

当センターでは 2015 年頃から，「計算・データ・学習」融合が，サイバー空間（仮想空間）とフィジカル空間（現実空間）を高度に融合したシステムを形成し，Society 5.0[3]が目指す人間中心の社会の実現に大きく貢献するものと考え，3 者の融合を実現するプラットフォームとして「『計算・データ・学習』融合スーパーコンピュータシステム」（通称 BDEC（Big Data & Extreme Computing））構築を目指して，様々な研究開発を進めてきました。

2021 年 5 月 14 日に運用を開始した新システム，「Wisteria/BDEC-01[4]」は BDEC システム構想に基づくスパコンシステムの第 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，総ピーク性能はそれぞれ 25.9 PFLOP（Odyssey），7.2 PFLOPS（Aquarius），合計 33.1 PFLOPS です。2021 年 11 月に発表された Top500 リスト[5]では，Odyssey が 17 位，Aquarius が 106 位となっております。現在は，Odyssey と Aquarius は別々に運用されておりますが，両者を連携させて「計算・データ・学習」融合を実現するためのソフトウェア群も，革新的ソフトウェア基盤 h3-Open-BDEC[6]の一環として整備されており，2022 年 4 月以降は多くのユーザーの皆様に「計算・データ・学習」融合を体験いただける見通しです。

さて，新しい仲間を迎える一方で去りゆく友もいます。Reedbush（データ解析・シミュレーション融合スーパーコンピュータシステム）[7]は当センターとしては初の GPU クラスタであり，また BDEC システム構想のプロトタイプとして 2016 年 7 月から運用を開始いたしましたが，2021 年 11 月 30 日を以て終了いたしました。Wisteria/BDEC-01 導入へ向けて様々な知見を得る

1 https://www.hpci-office.jp/
2 https://www.hpci-office.jp/pages/adoptionlist2021_25
3 https://www8.cao.go.jp/cstp/society5_0/
4 https://www.cc.u-tokyo.ac.jp/supercomputer/wisteria/service/
5 https://www.top500.org/
6 https://h3-open-bdec.cc.u-tokyo.ac.jp/
7 https://www.cc.u-tokyo.ac.jp/supercomputer/reedbush/service/

とともに，医療画像処理などこれまでとは違った分野のユーザー開拓など，当センターにとっては転回点ともいうべき時期に，重要な役割を果たしてくれました。

また，筑波大学と共同で運営する JCAHPC（最先端共同 HPC 基盤施設）による Oakforest-PACS（メニーコア型大規模スーパーコンピュータシステム，OFP）[8]は 2016 年 10 月に運用を開始いたしましたが，これも 2022 年 3 月末を以て終了します。OFP は国内最大級のシステムであり，特に 2019 年 8 月末の「京」運用終了後は，「富岳」が本格的に稼働を開始するまでの約 2 年間，National Flagship System の代役を果たし，我が国における計算科学コミュニティに多大な貢献をしてくれました。2021 年 11 月の Top500 でのランキングは 39 位で，引退には惜しいと言えるかも知れません。JCAHPC では引き続き筑波大学と共同で OFP の後継機（OFP-II）の検討を開始しており，2024 年 4 月の運用開始を予定しております。新しい仲間が加わるまで寂しくなりますが，しばらくお待ちください。

2021 年は「脱炭素化」という世界的潮流もありました。スパコン自体は脱炭素化に向けた様々な研究開発の取り組みに貢献していますが，消費電力を下げることも重要です。当面の性能要求を考慮すると，OFP-II 以降は GPU 等の加速器搭載ノードを中心とせざるを得ず，それを見据えて利用者の皆様には，加速器搭載システムへのアプリケーション移行に関するアンケートをお願いしているところです。2021 年 11 月 30 日で一旦締め切っていますが，継続的に受け付けておりますので，今からでも是非ご回答ください。

スーパーコンピュータの処理能力の向上に伴い，扱うデータ量も増加の一途をたどっておりますが，当センターでは従来ストレージは各システムに附属して導入され，各システムのストレージは独立でした。ただ。このような状況は，利用者に多大な不便を強いることになり，当センターの全スパコンシステムからアクセス可能な共通ストレージの導入が強く求められていました。このたび，OFP の運用終了とそれによるファイル移行の作業を契機に，各スパコンシステムからアクセスできる「大規模共通ストレージシステム（Ipomoea：サツマイモの学名）」の導入を決定いたしました。その第 1 号機である「Ipomoea-01（25PB）」は 2022 年 1 月末に運用を開始し，当面は OFP からのファイル移行に使用され，2022 年 6 月頃から一般に利用できるようにする予定です。3 年後の 2025 年には同程度の容量の 2 号機「Ipomoea-02」を導入予定です。2 つの共通ストレージシステムを並行して運用し，1 システム 5-6 年使用，3 年ごとに一方を更新というサイクルを定着させ，皆様により効率的にスパコンシステムをご利用いただければと考えております。

2015 年から「計算・データ・学習」融合という方針を掲げて 6 年あまり，2021 年は当センターにとっては「計算・データ・学習（Simulation+Data+Learning，S+D+L）」融合元年とも言うべき特筆すべき年となりました。Wisteria/BDEC-01 は「計算・データ・学習」融合を実現する，ヘテロジニアスなシステムとしては世界でも初めての画期的なスパコンです。繰り返しになりますが，準備も整いましたので，本年は是非「計算・データ・学習」融合により更なる新しい計算科学・計算工学の開拓に更に貢献したく，よろしくお願いいたします。

様々な問題の解決に当センターの「スパコン＋ストレージ」システム群（Oakbridge-CX，Wisteria/BDEC-01，Ipomoea-01）が役立てられるよう，利用者の皆さまとは一層緊密にご協力させていただければと思います。是非お気軽にご相談ください。

それでは，本年（「計算・データ・学習」融合 2 年）もよろしくお願いいたします。

[8] https://www.cc.u-tokyo.ac.jp/supercomputer/ofp/service/

センターから

サービス休止等のお知らせ

2022 年 1 月下旬からの計算機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
1 月 28 日（金）	9:00 ～ 22:00 までサービス休止	月末処理
2 月 25 日（金）	9:00 ～ 22:00 までサービス休止	月末処理
3 月 31 日（木）～ 4 月　4 日（月）	3/31 9:00 ～ 4/4 17:00 までサービス休止	年度末処理

・ 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
1 月 27 日（木）9:00 ～ 28 日（金）9:00 まで 2 月 24 日（木）9:00 ～ 25 日（金）9:00 まで 3 月 30 日（水）9:00 ～ 31 日（木）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
1 月 28 日（金）	9:00 ～ 20:00 までサービス休止	月末処理
2 月 25 日（金）	9:00 ～ 20:00 までサービス休止	月末処理
3 月 31 日（木）～ 4 月　4 日（月）	3/31 9:00 ～ 4/4 17:00 までサービス休止	年度末処理

・ 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金曜日） 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
1 月 27 日（木）9:00 ～ 28 日（金）9:00 まで 2 月 24 日（木）9:00 ～ 25 日（金）9:00 まで 3 月 30 日（水）9:00 ～ 31 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。

Oakforest-PACS スーパーコンピュータシステム【 3月31日 9:00 サービス終了 】

○ Oakforest-PACS 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
1月 26日（水）	9:00 ～ 22:00 までサービス休止	月末処理
2月 22日（火）	9:00 ～ 22:00 までサービス休止	月末処理
3月 31日（木）	9:00 すべてのサービスを終了	サービス終了

・ Oakforest-PACS スーパーコンピュータシステムは，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。

○ Oakforest-PACS 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
1 月 25 日（火）9:00 ～ 26 日（水）9:00 まで 2 月 21 日（月）9:00 ～ 22 日（火）9:00 まで 3 月 30 日（水）9:00 ～ 31 日（木）9:00 まで

・上記期間中，Oakforest-PACS の regular-flat，regular-cache，debug-flat，debug-cache，interactive-flat，interactive-cache 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，prepost キューは通常どおり利用できます。

【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，Oakbridge-CX 及び Oakforest-PACS 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

(2021.11.1 － 2021.12.31 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 Oakforest-PACS スーパーコンピュータシステム … 12/15

- プリポストノード 8 台（ofp11 - 18）を、1 月から実施するデータ移行用サーバとして使用するため運用から除外

2. ソフトウェア

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

- Odyssey (Wisteria-O)

富士通社製コンパイラ (Fortran77/90/95/2003/2008、C、C++)	1.2.34	(2021.12.17)

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) … なし

2.3 Red Hat Enterprise Linux 7, CentOS 7 (Oakforest-PACS) … なし

3. その他

3.1 Reedbush-H/L サービス終了について

Reedbush-H/L スーパーコンピュータシステムは 2021 年 11 月 30 日 9:00 をもってシステムを停止し、すべてのサービスを終了致しました。2020 年 6 月に終了した Reedbush-U と共に、Reedbush スーパーコンピュータシステムをご利用いただきありがとうございました。

3.2 Oakforest-PACS サービス終了について

Oakforest-PACS スーパーコンピュータシステムは 2022 年 3 月 31 日 9:00 をもってすべてのサービスを終了致します。詳細については Web ページ、メール、スーパーコンピューティングニュースをご参照ください。

Oakforest-PACS サービス終了にあたっては以下の点にご注意ください。

- サービス終了後のスーパーコンピュータのご利用につきましては Wisteria/BDEC-01、Oakbridge-CX をご検討ください。
- 一般利用にて Oakforest-PACS をご利用の方は「トークン移行」を行うことが可能です。Wisteria/BDEC-01、Oakbridge-CX への移行をご検討の利用者様につきましては「トークン移行」も併せてご参考ください。「トークン移行」についての詳細は Web ページ(https://www.cc.u-tokyo.ac.jp/guide/application/transfer_token.php)をご参照ください。
- Oakforest-PACS において、3 月 31 日 9:00 の時点にて、残トークンがあったとしてもこれ以降残トークンを消費することはできません。
- Oakforest-PACS の利用者様におきましては、2022 年に稼働する東京大学情報基盤センター「大規模共通ストレージシステム(Ipomoea-01)」へのファイル移行サービスについてのご案内を、最先端共同 HPC 基盤施設(JCAHPC)より別途メールで送らせていただいております。

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

スーパーコンピューティングチーム

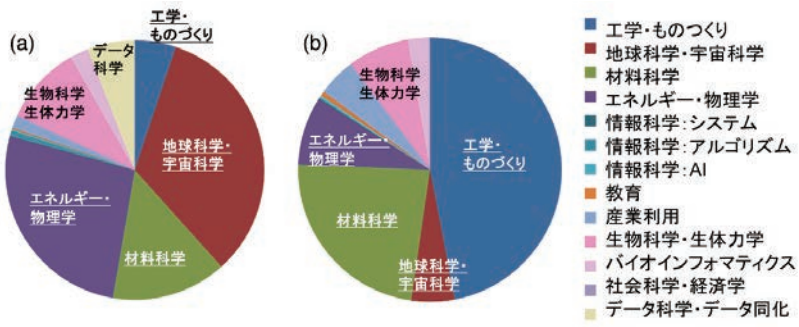
東京大学情報基盤センター(以下「当センター」)におきまして、当センターで運用する各スーパーコンピュータシステムからアクセス可能なストレージシステム：大規模共通ストレージシステム(第１世代) (Ipomoea-01)を稼働/運用いたします。システム稼働は 2022 年 1 月下旬からとなりますが、2022 月 3 月末に運用を終了する Oakforest-PACS (OFP) のデータ移行(希望者のみ)及びシステム調整作業を行い、2022 年 6 月から利用者様にご利用いただく予定です。ここでは、本システムの基本的な仕様並びに、概要についてお知らせいたします。なお、運用開始までに、予告なく運用仕様の変更を行う場合がありますので予めご了承ください。

1. 背景

当センターは 1965 年に東京大学大型計算機センターとして設立されて以来 50 年余り、全国共同利用施設、学際大規模情報基盤共同利用・共同研究拠点の中核拠点、革新的ハイパフォーマンス・コンピューティング・インフラ (HPCI) の構成機関として、国内外の産学官の各機関で実施されているスーパーコンピュータを使用した大規模シミュレーションによる計算科学・計算工学の研究の発展に多大な貢献をしてきました。2022 年 1 月現在、当センターでは 3 式 (「メニーコア型大規模スーパーコンピュータシステム (Oakforest-PACS (OFP))」、「大規模超並列スーパーコンピュータシステム (Oakbridge-CX (OBCX))」、「『計算・データ・学習』融合スーパーコンピュータシステム (Wisteria/BDEC-01)」) のスーパーコンピュータシステムを運用しており、総利用者数は学内外を合計して約 2,600 名以上となっております。各システムは、高い計算性能、ユーザーフレンドリなプログラム開発環境、安定した運用が利用者に高く評価されています。また、当センターのシステムは、ものづくり、地球・宇宙科学、物性科学などの計算科学・計算工学分野で広く利用されてきましたが、昨今はデータ科学、生物科学などより多様な分野で使用されるようになっています。

当センターの既存 3 システムはいずれも、「第三の科学 (The Third Pillar of Science)」と呼ばれる計算科学に加え、更にデータ科学、機械学習/人工知能を融合した新しい分野の開拓に資するものであり、サイバー空間（仮想空間）とフィジカル空間（現実空間）を高度に融合させたシステムにより、経済発展と社会的課題の解決を両立する、人間中心の社会 (Society) すなわち Society5.0 の実現に貢献することが期待されています。

これらスーパーコンピュータの処理能力の向上に伴い、扱うデータ量が増加の一途をたどっています。特に「計算・データ・学習」の融合を目指す新しい分野では、大量の観測データ、パラメータスタディの結果ファイルなどを処理する必要があります。当センターでは従来、ストレージは各システムに附属して導入され、各システムのストレージは独立しておりました。近年では当センターのシステム数も増加し、利用者も目的や手法に応じて複数のシステムを同時に利用する事例が増加しており、システムがリプレースされる場合には大量のデータをバックアップする必要がありました。個別のワークロードのデータ量も増加していることから、当センターの全システムからアクセス可能な共通ストレージの導入が強く求められている状況となっております。利用者に多大な不便を強いるこのような状況を考慮し、当センターでは、各システムからアクセスできる「大規模共通ストレージシステム(第 1 世代)」(Ipomoea-01) を導入することとしました。



実行ジョブノード時間の分野別比率 (2020 年度)
(a) Oakforest-PACS、(b) Oakbridge-CX

1. システム構成

1.1 ハードウェア構成

Ipomoea-01 のシステム構成は以下の通りです (図 1、表 1)。

図 1. システム構成図

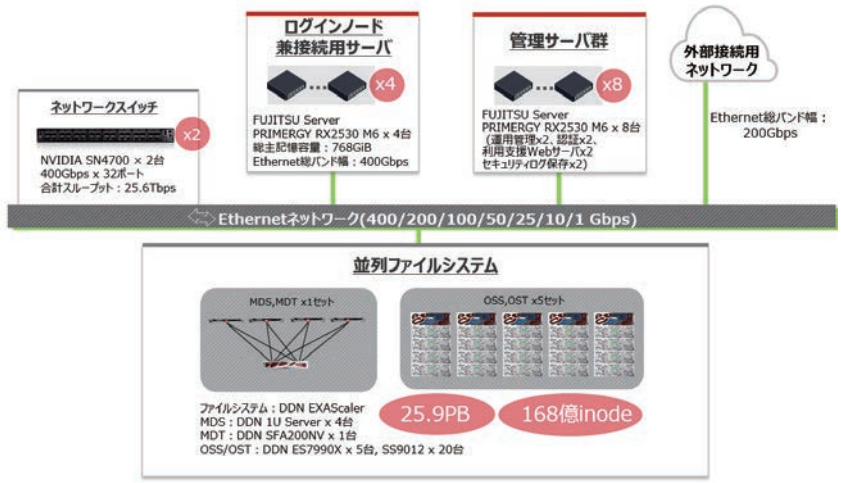


表 1. 並列ファイルシステム構成

項目		Ipomoea-01
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 x 4 台 + DDN SFA200NV x 1 台
	メタデータ格納デバイス	NVMe SSD 3.84 TB
	OSS + OST	DDN ES7990X x 5 台 + SS9012 x 20 台
	ファイルデータ格納デバイス	SAS-HDD 18 TB

1.2 ソフトウェア構成

Ipomoea-01 のソフトウェア構成(ログインノード)は、以下の通りです (表 2)。

表 2. ソフトウェア構成

※　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

	Ipomoea-01
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

2. サービス内容及び利用申込みについて

サービス内容やご利用になるための利用申込み方法の詳細は決定次第、当センター Web ページにてお知らせいたします。

2.1 利用申込みについて

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、利用申込み手続きが必要なケースがあります。

- Oakbridge-CX、Wisteria/BDEC-01 に利用者番号を有する場合
  利用申し込み手続きは不要です。ご利用の利用者番号で 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。
  - グループごとの領域に、登録されているスパコン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 %のディスク容量を無償で付与します。(トークン移行先のシステムを除く)
- Oakbridge-CX、Wisteria/BDEC-01 に利用者番号を持たない場合
  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。

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、スパコンシステムに利用者番号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。

2.2 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について、利用者様への運用を開始する 2022 年 6 月以降発生する予定です。

- Oakbridge-CX、Wisteria/BDEC-01 に利用者番号を有する場合
  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。**
- Oakbridge-CX、Wisteria/BDEC-01 に利用者番号を持たない場合
  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。

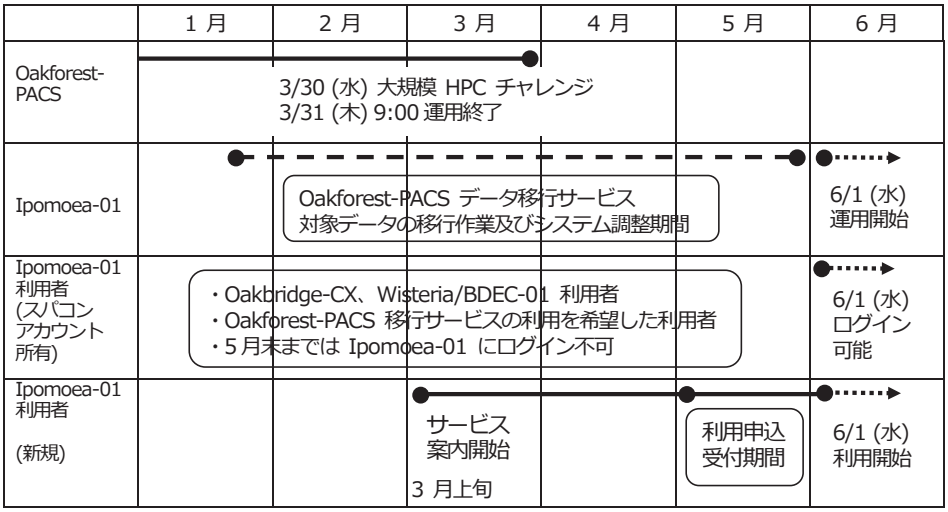
2.3 Oakforest-PACS からのデータ移行について

ファイル移行サービスについてのご案内を、最先端共同 HPC 基盤施設(JCAHPC)より別途メールで送らせていただいております。

2.4 運用開始までのスケジュールについて

Ipomoea-01 の運用開始までのスケジュールは以下の通りです (表 3)。

表 3. Ipomoea-01 運用開始までの流れ (予定)



2.5 運用開始後のサービスについて

2.5.1 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、ログインノードを複数台用意する予定です。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ーシステムと同様に、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 利用支援ポータルにてお知らせいたします。

2.5.2 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 2 の通りです（ただし、以下の図は、スーパーコンピューティングニュース原稿作成時のものです。今後の検討状況によっては、変更する場合もあります）。

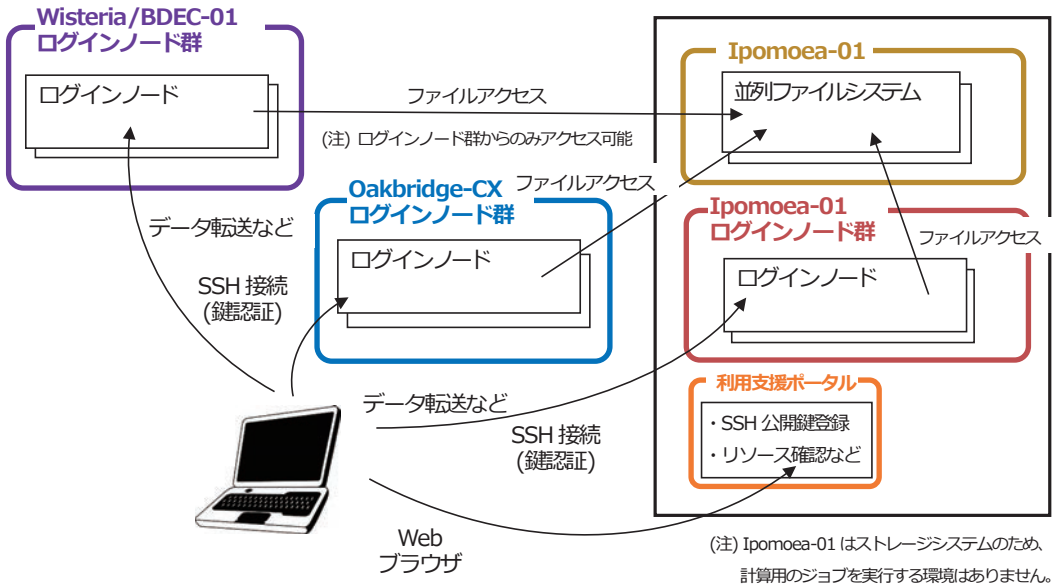


図2．システムのご利用イメージ

2.5.3 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、図 2、表 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、Oakbrige-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）が可能なように設定する予定です。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(利用者番号)：利用者ごとの領域(ホームディレクトリ)
- /work/グループ名(プロジェクトコード)：グループごとの領域(データ用ディレクトリ)

※ /home は主にホームディレクトリ用途のため、Ipomoea-01 新規利用者における申し込み時のディスク容量は /work に割り当てられます。

3. 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。

2022 年度の利用申込（新規・継続）について

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

2022年度のスーパーコンピュータシステム利用申込（新規・継続）は下記のとおり取り扱います[1]。利用申込の内容によって、利用申込期限が異なりますのでご注意ください。

なお、この情報は本原稿作成時点での予定ですので、変更する場合があります。最新の情報については、スーパーコンピューティング部門Webページ（https://www.cc.u-tokyo.ac.jp/）でご確認ください。

また、**2022年4月からの利用に関する「利用登録のお知らせ」の送付については、2022年3月末を予定しております**ので、予めご了解ください。

１．新規利用申込

Wisteria/BDEC-01、Oakbridge-CX のグループコース

2022年1月下旬にスーパーコンピューティング部門Webページ（https://www.cc.u-tokyo.ac.jp/）に2022年度版の利用申込サイトを公開します。スーパーコンピュータシステム利用規程等をよくお読みになり、利用申込を行ってください。

申込は随時受け付けますが、**2022年4月初めからのご利用を希望される場合は、利用申込期限までに手続きを行ってください。**

なお、**申込状況により利用のお断りもしくは希望セット数どおりの提供ができない場合があります。**

■利用申込期限：2022年2月11日（金）（早めに手続きを行ってください）

※2022年度より、パーソナルコースは廃止され、グループコースのみとなります。

[1] 企業、若手・女性、大規模 HPC チャレンジ、萌芽共同研究公募課題「All for HPC」、学際大規模情報基盤共同利用・共同研究拠点及び HPCI 等の公募制度による利用、講義・講習会等の教育利用及びトライアルユースを除きます。

２．継続利用申込

Wisteria/BDEC-01、Oakbridge-CXのグループコース

2022年1月下旬に代表者の方に2021年度の登録内容を記載した「継続手続きの案内」を送付します。2022年度も継続利用される場合は、案内に従い、利用申込期限までに手続きを行ってください。

なお、**申込状況により利用のお断りもしくは希望セット数どおりの提供ができない場合があります。**

また、提供セット数の決定にあたっては、2021年度利用実績、研究成果登録状況等を参考にさせていただく場合があります。

Oakforest-PACSにおきましては、2022年度3月を持ちまして**サービスを終了となります。**
Wisteria/BDEC-01、またはOakbridge-CXへトークン移行にてお申込みください。

■利用申込期限：2022年2月11日（金）（早めに手続きを行ってください）

※2022年度より、パーソナルコースは廃止され、グループコースのみとなります。

３．問い合わせ先

〒277-0882
千葉県柏市柏の葉6-2-3（東京大学情報基盤センター内）
東京大学情報システム部
情報戦略課研究支援チーム
E-mail：uketsuke@cc.u-tokyo.ac.jp

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

1. はじめに

Wisteria/BDEC-01、Oakbridge-CX、Oakforest-PACS 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占有利用ができる公募型プロジェクトです[1]。(※)

課題審査委員会による厳正な審査の結果、以下の課題を採択しましたのでお知らせいたします。

※ 新型コロナウイルス感染症拡大防止に配慮し、通常から一部条件、実施時間等を変更し実施しています。

- 実施時間は 8 時間（実施日当日 9:00～17:00）
- Oakforest-PACS スーパーコンピュータシステムにおいては、flat モード（4,200 ノード）、cache モード（3,200 ノード）の各設定の上限までの利用とします（各月 flat モード 1 件、cache モード 1 件の最大 2 件まで受入可能、ただし 1 グループで flat モード、cache モード両方利用することも可能）
- Wisteria/BDEC-01 スーパーコンピュータシステムにおいては、Wisteria-O（Odyssey）の 6,144 ノード（294,912 コア）又は Wisteria-A（Aquarius）の 36 ノード（288 基）、或いは両方の利用とします。

2. 採択課題

システム：Oakforest-PACS

募集期間：2021 年度 第 2 回再募集　2021 年 8 月 5 日～9 月 6 日

2 件の応募があり、以下の課題を採択しました。

採択課題一覧

課題名	並列多重格子法ソルバーの最適化および性能評価
代表者名 (所属)	中島　研吾（東京大学情報基盤センター）
連立一次方程式の解法の一種である多重格子法（multigrid method）は，反復回数が問題規模に依存しない，スケーラブルな解法であり，特に大規模問題に適した手法として知られているが，超並列環境下では性能低下が生じる可能性がある。提案者等はこれまで CGA（Coarse Grid Aggregation），hCGA（Hierarchical CGA），AM-hCGA（Adaptive Multilevel hCGA）法などの手法を提案し，MPI プロセス数が 105 以上の場合にも，スケーラビリティを保つことに成功している。また，OS 軽量カーネルである McKernel の適用により，超並列環境下で通信のオーバーヘッドを削減することによって，OFP 2,048 ノードを利用した場合，20%以上の性能向上が可能であることも明らかになっている。本研究では，並列多重格子法による三次元地下水シミュレーションプログラム pGW3D-FVM を対象として，SIMD ベクトル化に適した SELLC-s に基づく疎行列格納法，混合精度演算導入によるノード単体の高速化を図り，最大 4,096 ノードを使用して，McKernel 適用の効果を様々な設定の OpenMP/MPI ハイブリッド並列プログラミングモデルにおいて評価する。更に，現在開発中の動的ループスケジューリングに基づく通信と計算のオーバーラップによる新手法についても評価を実施する。	

[1]「大規模 HPC チャレンジ」
https://www.cc.u-tokyo.ac.jp/guide/hpc/

課題名	温水冷却方式の効率に関する定量的評価に向けて
代表者名 (所属)	庄司 文由 (理化学研究所・計算科学研究センター・運用技術部門)

近年、温水冷却技術は、HPC システムの冷却にかかるエネルギー効率を改善するために、多くの HPC センターおよびデータセンターで採用されている。温水冷却においては、CPU 等の冷水温度を高く設定することで、外気による自然な冷却が促されるため、冷凍機等を駆動するための電力が節約できる。

一方で、CPU 等のシリコンから構成される半導体は、高温で動作させればさせるほど、漏れ電流の増加により、消費電力が増加することが知られている。また、近年の CPU は、駆動温度や負荷が一定の水準を超えると、故障を回避するために自動的にクロック周波数や電圧を下げる機構を備えている。このため、温水冷却の効果を正しく評価するためには、冷却の電力を節約できるというポジティブな点に加え、上記のような消費電力の増加や、演算性能の低下のようなネガティブな点も等しく考慮に入れる必要がある。特にクロック周波数の低下による影響は、大規模なジョブでより深刻になると予想される。一般的に、CPU(プロセス)間で同期を取る際の性能は、最も遅い CPU(プロセス)に律速されるからである。以上を踏まえ、本研究では、単体および複数の CPU を使う様々なジョブを、異なる冷水温度で実行し、消費電力の増加および演算性能の低下を定量的に分析する。左記の分析に基づき、効率的な施設運用の実現に向けた、運用手順の確立を目指す。

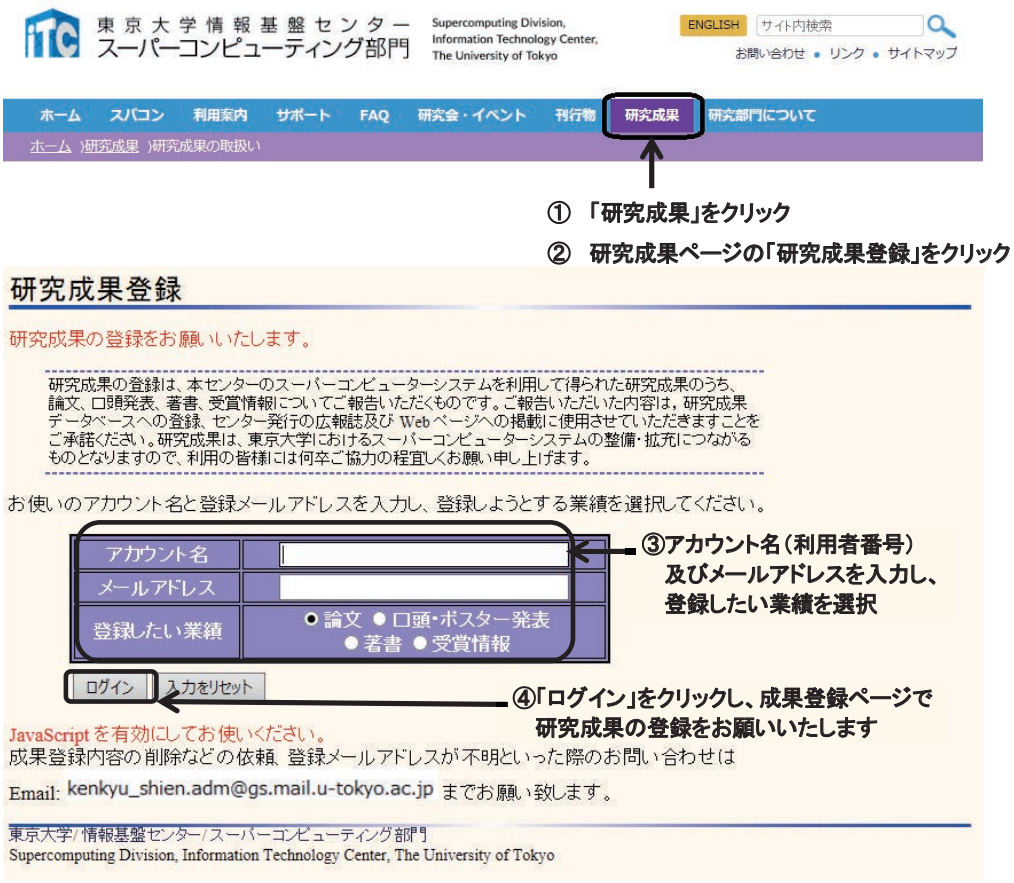
今回は、前回 (2019 年 10 月) および前々回 (2021 年 6 月) の大規模 HPC チャレンジで十分なデータが得られなかった複数 CPU を使うジョブのケースについて重点的に調査したい。前回も複数 CPU を使うジョブのケースの調査を行ったが、その際、並列ジョブを流すための CPU のグループ分けに LINPACK の単体 CPU 性能データを考慮せずに行ったため、グループ間での性能差がクリアに見えなかった。今回は、LINPACK の実行性能に基づいたグルーピングを実施することで、複数 CPU を使うジョブを実行した際の性能への影響を直接的かつ定量的に分析できると考えている。

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ー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、ご了解ください。

研究成果は、東京大学におけるスーパーコンピューターシステムの整備・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。

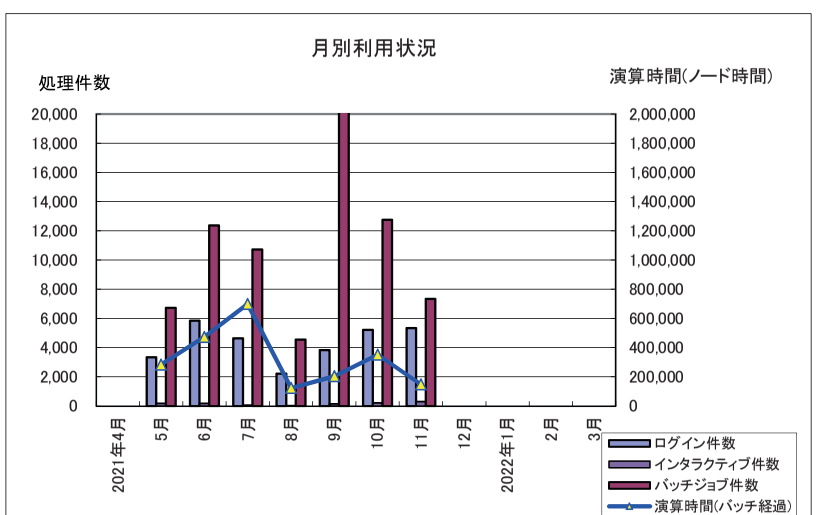


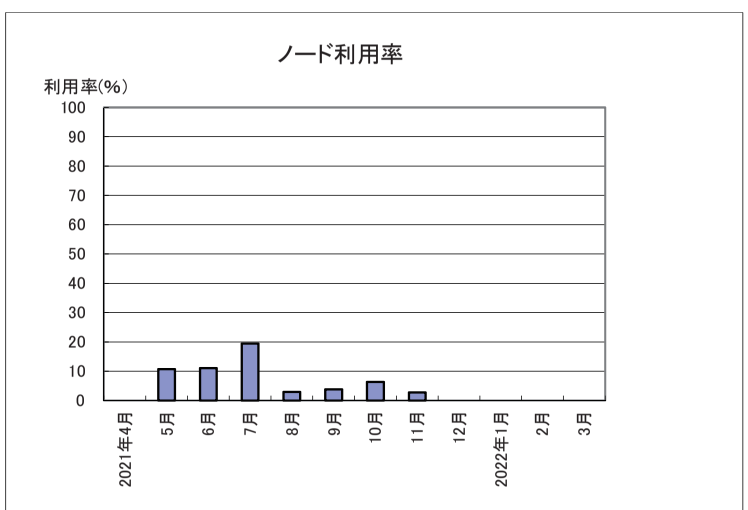
10・11月のジョブ統計

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		
2021年5月	860	206	3,348	54	188	6,718	13,623	148	66,121	697.32	40	35	282,715	822.0	10.7
6月	974	270	5,845	97	186	12,367	23,491	361	189,018	1,398.60	122	36	473,446	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	698,405	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
合計			30,436	707	1,153	74,896	145,737			11,859	1,142	303	2,281,151		

・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
・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ヶ月の稼動時間
・ノード利用率： サービスノードに対する利用比率。 計算式＝ノード利用数÷サービスノード数×100



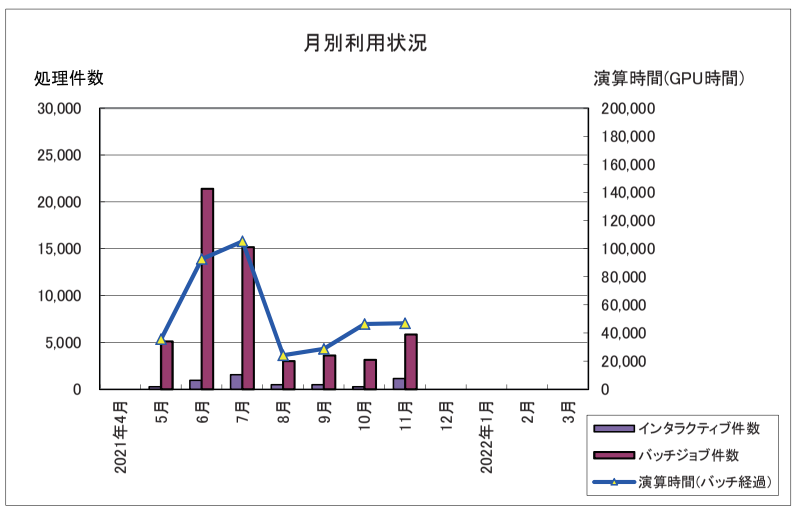


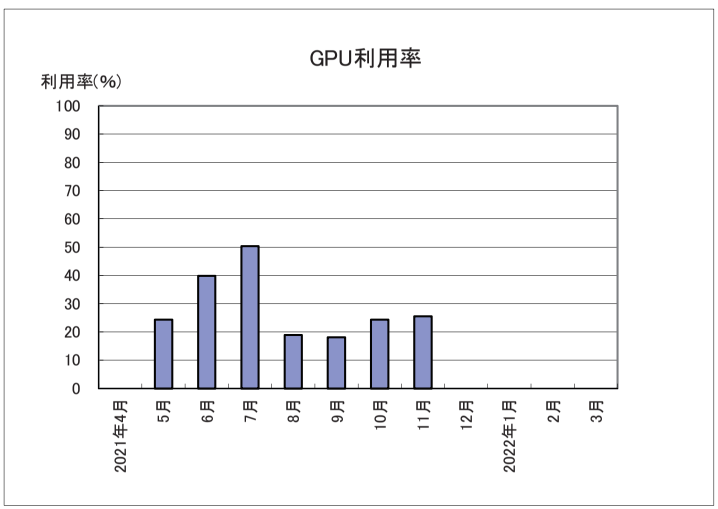
2. Wisteria/BDEC-01スーパーコンピュータシステム（Aquarius）ジョブ処理状況 （RedHat Enterprise Linux 8）

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		
2021年5月	252	5,095	158	35,645	87.5	24.3
6月	953	21,394	1,245	92,643	143.3	39.8
7月	1,540	15,155	1,531	105,239	180.9	50.3
8月	466	3,004	470	24,188	67.9	18.9
9月	480	3,599	358	28,702	65.1	18.1
10月	272	3,139	91	46,327	87.5	24.3
11月	1,122	5,824	1,107	46,895	91.9	25.5
合計	5,085	57,210	4,960	379,639		

・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Wisteira/BDEC-01(Odyssey) と共通。

・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



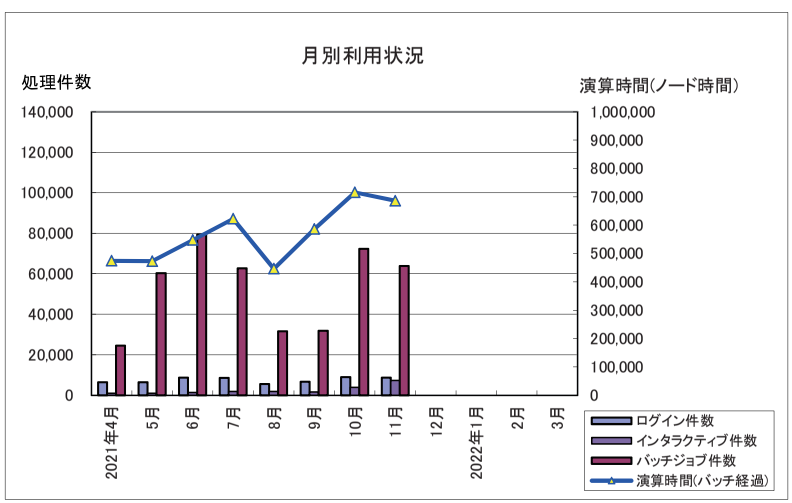


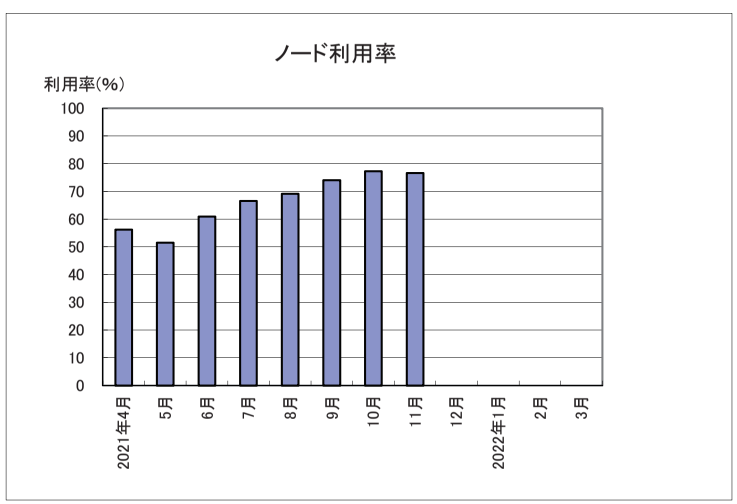
3. Oakbridge-CX スーパーコンピュータシステムジョブ処理状況（Red Hat Enterprise Linux 7、CentOS 7）

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		
2021年4月	1,070	323	6,405	2	927	24,498	27,351	913	1,339,482	1,242.42	0	385	474,088	769.1	56.2
5月	1,196	299	6,395	0	982	60,233	38,699	874	1,292,853	8,996.00	0	992	472,558	704.1	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	547,424	832.8	60.9
7月	1,132	366	8,545	42	1,860	62,733	57,857	1,017	1,430,169	16,807.20	2	674	622,202	909.5	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	446,153	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	586,073	1,011.9	74.0
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	715,570	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	686,231	1,047.8	76.6
2020年11月	1,241	268	13,169	13	2,707	47,307	62,082	1,172	1,878,875	15,583.41	17	980	716,567	1,086.7	79.9
12月	1,214	241	11,452	56	1,475	75,087	43,511	1,191	1,872,689	9,074.40	253	860	768,743	1,127.9	82.9
2021年1月	1,105	215	10,238	49	1,601	64,441	53,307	1,021	1,891,719	3,322.22	253	982	754,704	1,110.5	81.7
2月	1,110	161	7,326	117	2,215	40,336	28,469	1,049	1,963,710	2,864.16	398	893	591,306	941.2	69.2
3月	1,080	170	7,803	67	1,373	63,001	35,810	1,115	1,502,680	2,660.82	387	671	591,775	854.0	62.8
合計			96,659	481	26,456	669,171	485,965			61,335	1,302	9,868	7,256,827		

・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
・2020年11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ヶ月の稼動時間
・ノード利用率： サービスノードに対する利用比率。 計算式＝ノード利用数÷サービスノード数×100





4. Oakforest-PACSスーパーコンピュータシステムジョブ処理状況 （RedHat 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）			平均ﾉｰﾄﾞ利用数（ノード）	ﾉｰﾄﾞ利用率（%）
			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		
2021年4月	1,690	450	8,322	128	181	68,373	47,950	2,618	6,233,823	3,528.98	310	65	2,731,333	4,117.0	50.2
5月	1,732	347	8,252	192	288	63,829	63,997	2,216	4,729,782	2,036.24	411	78	2,977,248	4,089.8	49.8
6月	1,397	378	9,348	274	511	59,097	66,215	2,236	4,830,259	2,088.30	470	92	3,179,605	4,557.0	55.5
7月	1,329	380	8,428	193	415	69,240	74,087	2,199	4,947,221	2,050.97	420	203	3,574,862	4,895.8	59.6
8月	1,336	308	5,994	189	237	33,416	28,905	2,177	5,049,162	5,565.78	523	160	2,981,494	5,366.4	65.4
9月	1,373	319	5,891	184	171	31,112	38,671	2,246	5,093,072	1,940.11	430	108	2,547,260	4,011.9	48.9
10月	1,414	329	6,956	170	113	28,226	59,362	2,212	5,137,733	1,148.12	363	45	2,165,785	2,978.2	36.3
11月	1,429	328	6,928	249	73	79,672	46,522	2,226	5,254,149	9,648.32	372	43	2,509,969	3,571.9	43.5
2020年11月	1,822	461	13,149	251	256	65,413	74,790	2,535	6,618,475	3,297.48	516	125	4,468,068	6,400.5	78.0
12月	1,789	471	13,407	419	375	99,699	66,020	2,648	6,634,466	6,273.23	630	149	4,837,811	6,633.6	80.8
2021年1月	1,804	475	12,636	495	556	210,638	84,101	2,687	6,512,605	3,174.92	787	233	5,112,137	6,993.7	85.2
2月	1,786	431	11,384	314	282	79,481	78,257	2,787	6,581,603	5,791.25	546	197	4,733,131	7,166.3	87.3
3月	1,727	472	12,986	384	302	271,840	102,642	2,760	6,629,043	6,329.17	706	174	5,484,594	7,523.7	91.7
合計			110,532	3,191	3,504	1,094,623	756,729			49,575	5,968	1,547	42,835,229		

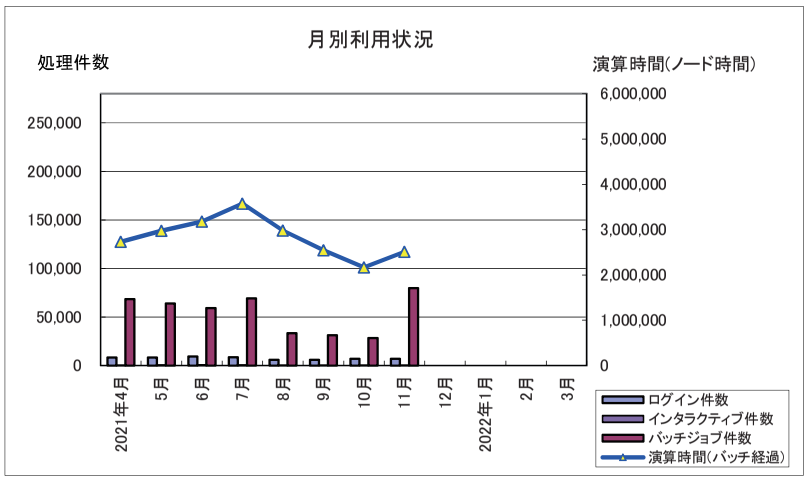
・接続時間： ログイン時間の累計

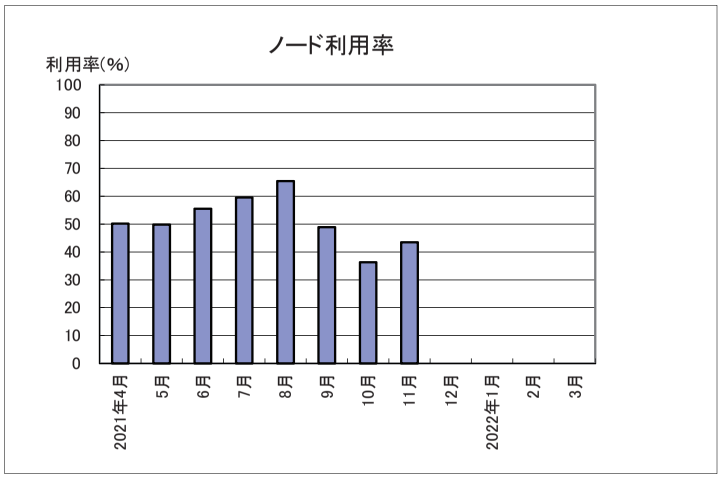
・ログイン（実CPU）： コア時間単位

・2020年9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＝ノード利用率×総ノード数(8208)

・ノード利用率： サービスノードに対する利用比率。 計算式＝利用ノード時間÷サービスノード時間×100

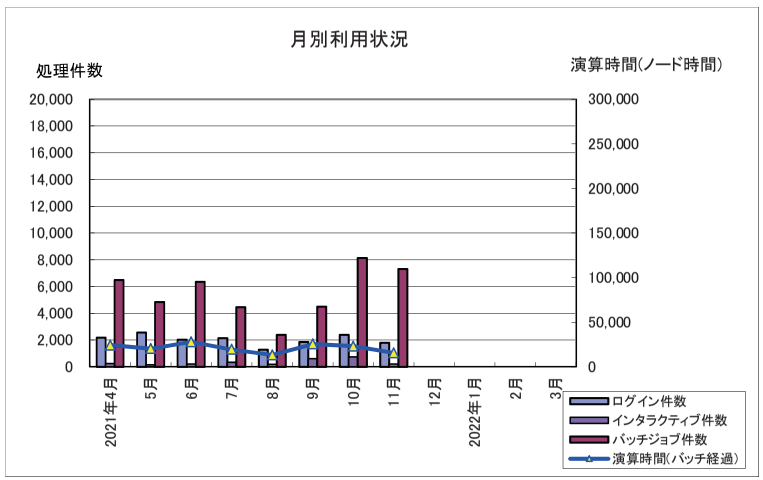


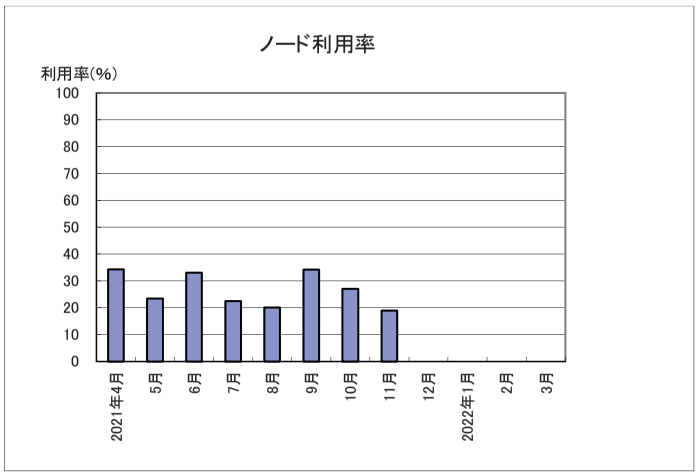


5. Reedbushスーパーコンピュータシステム（Reedbush-H）ジョブ処理状況 （RedHat Enterprise Linux 7）

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	/lustre		プリポスト	インタラクティブジョブ	バッチジョブ		
2021年4月	986	127	2,183	19	253	6,489	3,140	146	192,825	12.31	2	284	24,386	41.1	34.3
5月	644	116	2,571	0	149	4,839	3,857	134	203,491	13.35	0	179	20,465	28.0	23.4
6月	592	102	2,022	0	200	6,355	3,567	135	200,432	2.60	0	200	27,992	39.6	33.0
7月	587	111	2,149	2	348	4,452	3,564	142	210,761	4.95	3	285	19,598	27.0	22.5
8月	585	91	1,285	1	195	2,384	2,378	137	218,311	5.10	2	200	13,374	24.2	20.1
9月	610	97	1,862	0	608	4,502	3,302	136	222,755	2.59	0	635	25,653	41.0	34.2
10月	626	91	2,396	0	745	8,134	3,748	140	216,528	12	0	712	23,171	32.4	27.0
11月	631	112	1,797	0	213	7,316	2,862	139	197,530	5.66	0	244	15,757	22.7	18.9
2021年11月	1,070	188	4,943	0	717	13,502	6,705	168	337,922	7.48	0	675	28,402	40.8	34.0
12月	1,094	236	5,252	0	1,545	16,439	8,449	180	349,615	6.64	0	1,979	47,463	67.1	55.9
2021年1月	1,112	195	4,585	0	1,010	12,940	7,523	181	356,486	7.35	0	1,173	44,632	62.2	51.8
2月	1,057	188	3,758	0	809	12,800	5,831	188	354,254	16.00	0	954	56,532	86.5	72.1
3月	1,056	176	4,140	0	776	7,345	5,273	189	316,348	13.20	0	920	73,742	102.4	85.4
合計			34,000	22	6,851	93,995	53,494			92	7	7,765	392,765		

・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
・2020年11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ヶ月の稼動時間
・ノード利用率： サービスノードに対する利用比率。 計算式＝ノード利用数÷サービスノード数×100





6. Reedbushスーパーコンピュータシステム（Reedbush-L）ジョブ処理状況 （RedHat Enterprise Linux 7）

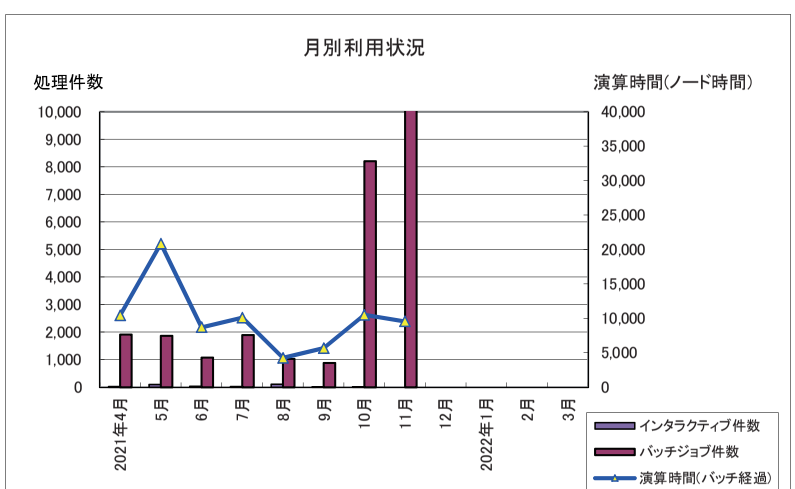
年月	処理件数		演算時間 [ノード時間](経過時間)		平均ﾉｰﾄﾞ利用数（ノード）	ﾉｰﾄﾞ利用率（%）
	インタラクティブジョブ	バッチジョブ	インタラクティブジョブ	バッチジョブ		
2021年4月	19	1,913	3	10,413	17.4	32.8
5月	95	1,867	45	20,833	28.3	53.5
6月	34	1,075	70	8,689	12.3	23.2
7月	25	1,887	44	10,081	13.8	26.0
8月	105	1,031	87	4,249	7.7	14.6
9月	1	874	0	5,657	8.8	16.7
10月	15	8,203	30	10,484	14.3	26.9
11月	0	14,338	0	9,546	13.5	23.8
2020年11月	56	1,695	25	24,476	34.4	63.7
12月	105	2,734	106	21,208	28.9	53.6
2021年1月	100	3,171	46	26,404	35.9	66.7
2月	56	2,657	211	28,072	41.1	74.7
3月	16	727	24	34,927	47.9	87.2
合計	571	40,477	666	190,563		

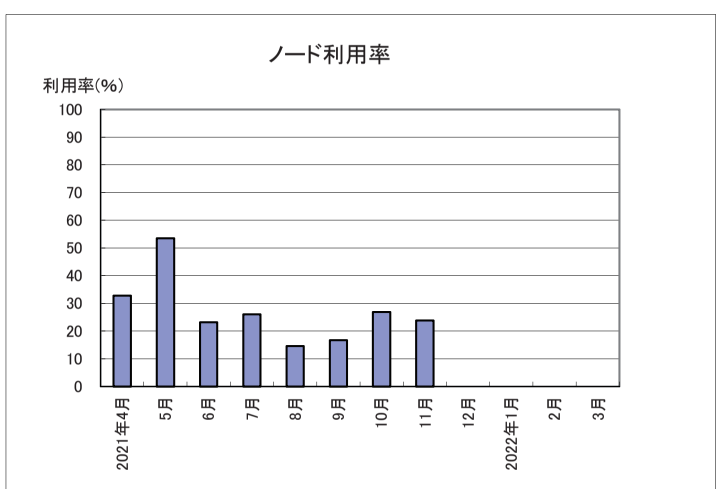
・登録者数、実利用者数、ログイン件数、接続時間、ファイル使用量、ログイン（実CPU）はReedbush-Uと共通。

・2020年11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ヶ月の稼動時間

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





SC21 参加報告

星野　哲也
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は、2021 年 11 月 15 日から 18 日までの間、オンラインと現地のハイブリッド開催された SC21（The International Conference for High Performance Computing, Networking, Storage, and Analysis）に参加し、研究展示を行いました。本稿では、SC21 に参加した東京大学情報基盤センター教職員が参加した中から気になったことを記します。

SC21 について

SC21 は高性能計算(HPC)分野に於いて最大級の国際会議であるとともに、様々な情報技術関連企業や研究所・大学等の技術展示会としても知られています。今回の開催地はミズーリ州セントルイスであり、オンラインとのハイブリッド開催を行いました。新型コロナウィルスの感染拡大状況を鑑み、東京大学情報基盤センターは例年より簡易な無人のブース出店を行いました。教職員の派遣は行わなかったため、SC の目玉イベントでもある、大ブースでの技術展示が見られず、寂しい印象がありました。また時差の都合上、日本からのリアルタイムでの参加はなかなか難しいものでしたが、発表の様子がムービー形式で後から視聴できるよう配慮されていたため、普段の SC なら並列進行故に見ることのできない同時間帯の発表も見ることができました。

各種ランキングについて

毎年のSCではスーパーコンピュータの性能に関する様々なランキングが更新されます。最も有名なスパコンランキングであるTop500、スパコンの電力性能を競うGreen500、実際のアプリケーションに近いと言われるHPCGや、スパコンのIO性能を競うIO-500などがあります。本稿ではTop500から見える全体的な傾向について述べます。

Top500（http://www.top500.org/）は世界のスーパーコンピュータの性能を LINPACK という係数行列が密の連立一次元方程式を解くベンチマークの処理速度によって競うものです。1993 年の開始以来、6 月にヨーロッパで行われる会議である ISC と、本会議 SC にて年 2 回の更新を続けています。

今回のランキングにおいては、2020 年 6 月のランキングから 4 回連続で理化学研究所

の計算科学研究センターが保有する富岳スーパーコンピュータが引き続きトップとなりました。富岳はA64FXと呼ばれる富士通が設計したARMアーキテクチャのプロセッサを158,976ノード搭載したスーパーコンピュータです。東京大学情報基盤センターに設置されたWisteria/BDEC-01のOdysseyノード群は富岳と同じA64FXを搭載しており、前回の6月のリストの13位から4ランクダウンし、今回は17位にランクインしました。

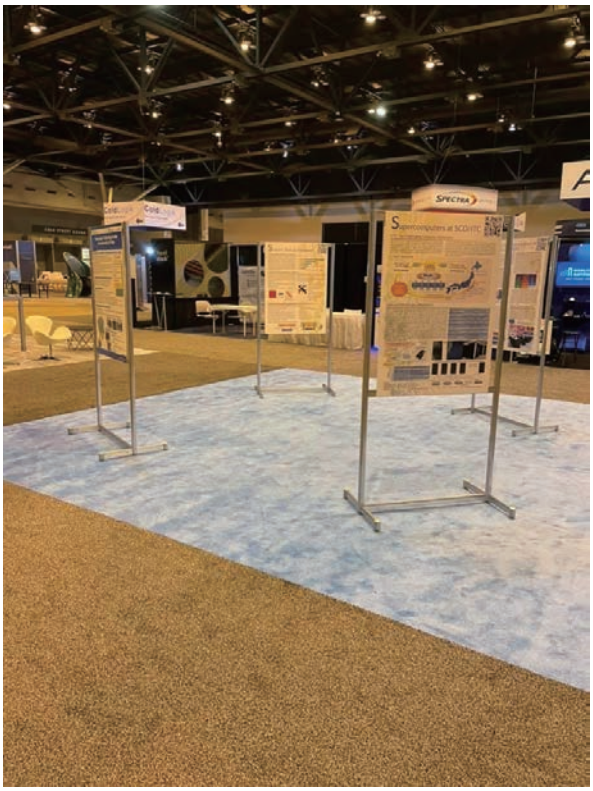
表1のRmax/Rpeakは実行効率と呼ばれる指標で、実際に得られた性能を理論性能で割ったものであり、オリジナルのTop500から筆者が追加した指標です。Rmax/Rpeakの値は高い程良いのですが、この値を高めるのはコア数が増加するほど難しくなります。富岳は763万ものコアを搭載しながらも82%の高い実行効率であり、同じくA64FXを搭載するOdysseyノード軍は85%でした。

Top10にランクインした他のシステムを見ると、6月のランキングから新たに加わったのは10位に入ったVoyager-EUS2のみでした。Voyager-EUS2はGPUを搭載したシステムであり、NVIDIAの最新のGPUであるA100 GPUを搭載しています。それより目を引くのは、Voyager-EUS2が民間会社であるMicrosoftが所有するスパコンであるということです。大きなスパコンを設置するのは主に研究機関でしたが、6位のSeleneもNVIDIA社が導入したスパコンであり、民間会社が大規模なスパコンを導入することがトレンドとなっています。これらのスパコンと同様にA100 GPUを搭載するWisteria/BDEC-01のAquariusノード群は、前回の93位から106位まで順位を下げました。

他に本センターが運用するスパコンでは、Oakforest-PACSがTop500の39位となり、Oakbridge-CXが110位となりました。

東京大学情報基盤センターによる展示

東京大学情報基盤センターは昨年同様、「ITC/JCAHPC, The University of Tokyo」の名義によるブース展示を行いました。筑波大学計算科学研究センターと共同で設立した最先端共同HPC基盤施設をJCAHPCと呼び、2016年12月より共同でOakforest-PACSスーパーコンピュータの運用を行なっています。例年は研究事例を紹介するポスター展示や、ブースでのプレゼンテーションが恒例となっておりましたが、今回は無人によるブース展示であったため、例年と比べると簡易なブース展示となりました。



東大ブースの様子（筑波大学教授の朴先生提供）

展示したポスターと同様のものを以下のリンクにも掲載しております。
https://www.cc.u-tokyo.ac.jp/public/sc21.php

Wisteria/BDEC-01 利用事例（2）

NVIDIA A100 と CUDA 11 の新機能

三木 洋平

東京大学情報基盤センター

1. はじめに

東京大学情報基盤センター（以下，本センター）が運用する「計算・データ・学習」融合スーパーコンピュータシステム（Wisteria/BDEC-01）のデータ・学習ノード群（Aquarius または Wisteria-A）には，NVIDIA 社の最新の GPU である NVIDIA A100 Tensor コア GPU（以下，A100）が全 360 基（ノード当たり 8 基，全 45 ノード）搭載されている[1]．A100 は，2021 年 11 月 30 日に運用を終了した Reedbush-H/L システムに搭載されていた NVIDIA Tesla P100（以下，P100）から性能・機能ともに強化された GPU である．「Wisteria/BDEC-01 利用事例」の第 2 回にあたる本稿では， A100 や CUDA 11 で導入された新機能とその使い方について紹介し，Reedbush-H/L から Aquarius への移行を支援する．

2. P100・V100・A100 の性能比較

本節では，過去の GPU と A100 の仕様を比較することで A100 の特徴を概観する．本センターが運用する GPU スパコンには搭載されなかったが，A100 と P100 の間には NVIDIA V100（以下，V100）があるため，この V100 も交えて表 1 に P100，V100，A100 のハードウェア観点での比較を示す[2， 3， 4]．各世代の GPU ともに複数の製品があるが，P100 および A100 については本センターが運用するシステムに搭載されている製品，V100 についてはこれに対応する型番の GPU の仕様を示した．

表 1：各 GPU の比較

	P100 (SXM)	V100 (SXM)	A100 (SXM, HBM2)
CUDA コア数	3584 (=56×64)	5120 (=80×64)	6912 (=108×64)
GPU Boost Clock	1.48 GHz	1.53 GHz	1.41 GHz
倍精度理論ピーク	5.3 TFlop/s	7.8 TFlop/s	9.7 (19.5) TFlop/s [1]
単精度理論ピーク	10.6 TFlop/s	15.7 TFlop/s	19.5 TFlop/s
半精度理論ピーク	21.2 TFlop/s	125 TFlop/s [2]	312 TFlop/s [3]
メモリ容量	16 GB (HBM2)	16 GB (HBM2)	40 GB (HBM2)
メモリ帯域幅	732 GB/s	900 GB/s	1555 GB/s
L2 キャッシュ容量	4 MB	6 MB	40 MB

[1] カッコ内は Tensor コアを使用した時の理論ピーク性能．
[2] Tensor コアの理論ピーク性能．
[3] Tensor コアの理論ピーク性能．

動作周波数については大きな違いがなく，Streaming Multiprocessor（SM）数が 56，80，108 とおよそ$\sqrt{2}$倍ずつ増えてきたことが性能向上の要因である．また，Volta 世代において導入された Tensor コアはもともと半精度の行列積和算のみを実行する演算機であったが，A100 においては倍精度の行列積和算にも対応したため理論ピーク性能を大きく上げている．さらに Tensor コアは倍精度，半精度に加えて TensorFloat32[4] に対応し，また非ゼロ要素だけを取り出して計算することで最大 2 倍の高速化を実現する Sparsity を導入するなど主に深層学習を意識した強化がなされている．ただし行列の積和算の専用演算機であるため全てのアプリケーションが恩恵を受けられるわけではないこともあり，本稿では Tensor コアについてはこれ以上掘り下げない．

メモリ性能については容量・バンド幅ともに増加してきているが，A100 における増強が大きい．特に L2 キャッシュが 40 MB に増量された恩恵は多くのアプリケーションが受けられるものである．

3. Volta 世代において導入された機能の復習

GPU の構成および CUDA のプログラミングモデルは，V100 および CUDA 9 において大きな変更が加えられ，最新の A100 および CUDA 11 はこの変更点をそのまま引き継いでいる．Reedbush-H/L から Aquarius へと直接移行するユーザはこうした変更点を見落としがちであるので，ここでまとめておく（V100 以降の GPU を活用したことのあるユーザにとっては既知の情報である）．

最大の変更点は“Independent Thread Scheduling”という動作モデルの導入である．P100 に代表される Pascal 世代以前の GPU においては，ワープを構成する 32 スレッドの演算は同時に実

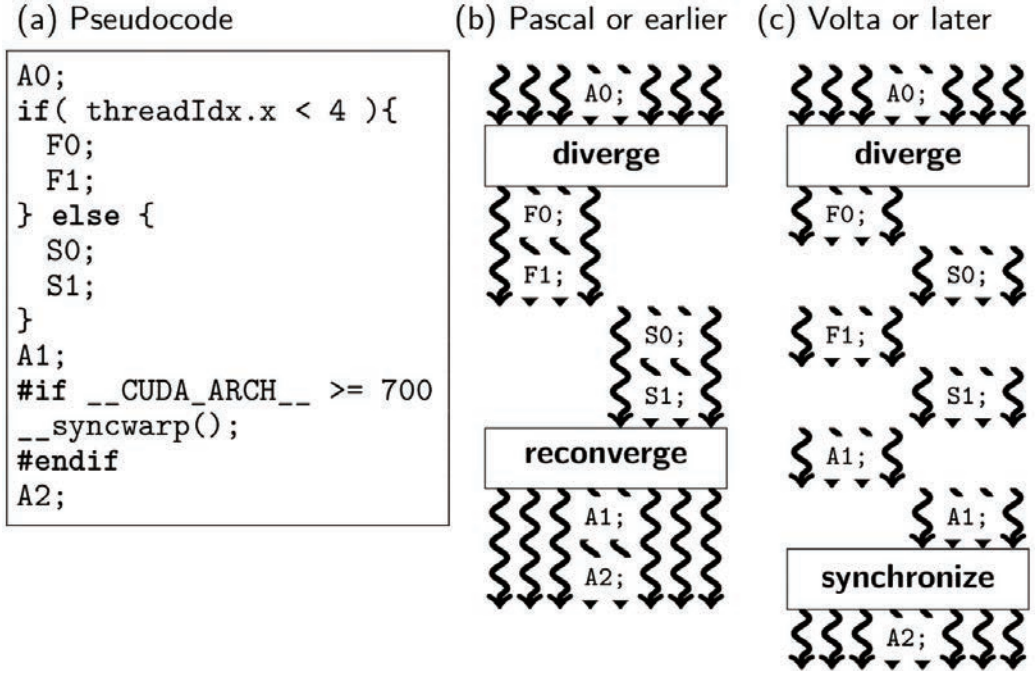


図 1　Independent Thread Scheduling の有無による動作モデルの違い．(a) 条件分岐を含んだコードの例．(b) Pascal 世代以前の GPU 上での動作モデル．(c) Volta 世代以降の GPU 上での動作モデル．

[4] 符合ビット 1，FP16 と同じ仮数部 10 ビットと FP32 と同じ指数部 8 ビットを持つ合計 19 ビットからなるデータ型であり，FP19 と呼ぶ方が実態を表している．

行されていたため，同期が必要な処理をワープ内に閉じ込めるよう実装した際には，明示的な同期命令を発行する必要がなかった．これに対して，Independent Thread Scheduling が導入された V100 や A100 においてはワープ内の 32 スレッドの演算が同時に実行されるという保証はなくなっており，ワープ内の 32 スレッドを同期する__syncwarp()，Cooperative Groups のタイル同期といった明示的な同期処理を行う必要がある（図 1）．特にワープ内の暗黙同期がなくなった影響として，if 文の実行などによってワープ分裂が生じた後には，明示的な同期処理を実行するまではワープが分裂したまま動作し続けるという点に注意が必要である．これは図 1b においては 1 回だけ実行されていた A1 が，図 1c においては 2 回実行されているという動作パターンである．図 1a において__syncwarp()を挿入する位置が A1 の後になってしまっているという実装ミスによって引き起こされる性能低下を回避するにも，慎重なコードの改修が必要となる．一方で，こうしたコードの書き換えが困難なユーザのための回避策も提供されている．通常 A100 向けに CUDA コードをコンパイルする際には「-gencode arch=compute_80,code=sm_80」を指定するが，この際に「-gencode arch=compute_60,code=sm_80」を指定することで Pascal 世代以前の動作モードを使用することができる（ただし，あくまで回避策なので推奨はしないとのことである）．

また P100 までは整数演算は CUDA コアにおいて実行されていたが，V100 からは整数演算用のコアが追加された．これによってインデックス計算などが浮動小数点演算と同時に実行される場合もあり，理論ピーク性能比以上の性能向上が実現されるアプリケーションもある[5]．

4. A100/CUDA 11 で導入された新機能とその使い方

本節では，A100 および CUDA 11 において導入された新機能について紹介する．CUDA 11 は，A100 対応がなされたバージョンの CUDA である[6]．代表的な新機能としては，Asynchronous copy/barrier，L2 cache residency control，Warp-wide reduction があげられる．

CUDA 11 において，グローバルメモリからシェアードメモリへと（レジスタを介さずに）直接データをコピーできる memcpy_async()が導入された．本命令は非同期実行されるため，データ転送と演算を同時に実行することで一方の実行時間を隠蔽できる．また，A100 上ではハードウェア的な加速が効くという利点もある．使い方としては，cuda/pipeline や cuda/barrier をインクルードして libcu++経由で使うこともできるが，本稿では実装が簡単な Cooperative Groups 経由での実装例を紹介しておく（図 2）．Svedin らによってマイクロベンチマークが行われており，pipeline API が barrier API よりも高速であること，メモリ律速な問題では性能が向上すること，演算律速な問題では性能が低下することが報告されている[7]．

```
#include <cooperative_groups.h>
#include <cooperative_groups/memcpy_async.h>
using namespace cooperative_groups;

// デバイス関数内での実装：
  thread_block_tile<TSUB> tile = tiled_partition<TSUB>(this_thread_block());
  // TSUB は，この関数内で memcpy_async に関与するスレッド数（この書き方の場合は32以下）

  cooperative_groups::memcpy_async(tile, &shared_mem, &global_mem, sizeof(uint) * num);
  // 何か裏で回したい処理があれば，ここに書く
  cooperative_groups::wait(tile);// ここでコピーの完了待ちをする
```

図 2　memcpy_async() の実装例．

A100においてはL2キャッシュの容量が40 MBと大幅に増量されたが，グローバルメモリ上の単一・連続なデータ領域をL2キャッシュ上に置き続けることもできるようになった．この際，L2キャッシュ容量の1/16である2.5 MB単位での調整が可能であるが，CUDAストリームごとに1つのデータ領域しか指定できないという制約がある．実装は図3のようになる．

```
cudaStreamAttrValue attr;

attr.accessPolicyWindow.base_ptr = /* beginning of range in global memory */ ;
attr.accessPolicyWindow.num_bytes = /* number of bytes in range */ ;

// hitRatio causes the hardware to select the memory window to designate as per
sistent in the area set-aside in L2
attr.accessPolicyWindow.hitRatio = /* Hint for cache hit ratio: 1.0で良い */

attr.accessPolicyWindow.hitProp = cudaAccessPropertyPersisting;
attr.accessPolicyWindow.missProp = cudaAccessPropertyStreaming;

cudaStreamSetAttribute(stream, cudaStreamAttributeAccessPolicyWindow, &attr);
```

図 3　L2 cache residency control の実装例.

ワープ内のリダクション演算を高速に計算できる__reduce_*_sync()命令が導入された．本命令は「-gencode arch=compute_80,code=sm_80」をつけてコンパイルする必要があるので，暗黙の同期との併用はできない．提供されている演算はadd，min，max，and，or，xorの6種類であり，32ビット整数型にのみ対応している（and，or，xorは符号なし整数型のみに対応）．前述の通り浮動小数点数には対応していないが，minおよびmaxについては大小関係を保存した上で適切な符号なし整数型へと変換することで流用することは可能である[8]．

本節の最後に，A100において顕在化したシェアードメモリ使用時の制限事項をリマインドしておく．デバイス関数内でシェアードメモリを静的に確保する際には，ブロックあたり48 KBが上限の容量である．過去のGPU,例えばV100においてはSMあたり96 KBのシェアードメモリが使用可能であったが，標準的にはSMあたりに複数のブロックを割り当てるため，48 KBの制限は実質的には存在しないと言えた．A100においてはSMあたり164 KB [5]までシェアードメモリを使えるため，例えばSMあたりのブロック数が2であれば静的な確保はできず，この場合には動的な確保が必要となる．

4. 重力ツリーコードGOTHICでの例

今まで紹介してきたA100およびCUDA 11の新機能を実アプリケーションに適用し，A100向けに最適化した事例を紹介しておく．ここでは，宇宙物理学の研究に用いるために開発されたN体シミュレーションコードGOTHIC［5，8，9］を対象とする．

GOTHICにおいては，遠方粒子からの重力を計算する際に多重極展開を用いて演算量を削減するツリー法，軌道進化の遅い粒子に関する重力計算・軌道積分を間引く階層化時間刻み法を採用しており，ほとんどの浮動小数点演算は単精度で実行されている．また，重力計算だけではなくツリー構造の構築などの処理もCUDA C/C++を用いてGPU化されている．

GOTHICのA100向け最適化としては，本稿で紹介した各種機能を使用するかどうかや関連する

[5] 1ブロックあたりでは163 KBまでのシェアードメモリが使用可能である．

パラメータの調整（例えば，L2 キャッシュにデータを固定する際の容量）が必要となる．このためには各パターンでの性能測定を実施し，最も高速なパラメータセットを抽出すれば良い．ただし重力ツリーコードの演算性能には用いる粒子分布に対する依存性もあるため，実際の宇宙物理学の研究で用いられる現実的な状況設定を準備しなければならない．そこで今回はアンドロメダ銀河を模した準力学平衡状態にある粒子分布を初期条件生成コード MAGI [10]によって生成した．また GOTHIC は階層化時間刻み法を採用しており重力計算に要する時間がステップごとに 2 桁以上変動するため，4096 ステップに渡る N 体シミュレーションを実行し，その全体の実行時間を測定することとした．また，この際の粒子数は$N = 2^{23} = 8388608$とした．

A100/CUDA 11 において導入された新機能としては，memcpy_async()については使わない方が高速であった．これは，演算律速な問題においては性能が低下する傾向にあるという Svedin らの報告[7]とも一致する．L2 キャッシュへのデータ固定についてはデータサイズに対する依存性がほぼなかったが，Independent Thread Scheduling 有効時（A100 のデフォルト，compute_80 を指定）においては粒子データを 40 MB 固定した場合，無効化時（compute_60 を指定した時）にはデータ固定を行わない場合が最も速かった．本機能もメモリ律速な問題に効果的と考えられるため，演算律速なアプリケーションである GOTHIC において大きな効果がなかったことは不思議ではない．最後にワープ内のリダクション命令については Independent Thread Scheduling 有効時のみしか使えないが，本命令を使用した方が高速になることが確認できた．また，CUDA のバージョンを変えた実験も行った結果，CUDA 11.2 よりも CUDA 11.1 の方が高速であったため，GOTHIC については CUDA 11.1 環境を用いて動作させることとした．

以下では，Aquarius 上での測定結果を紹介する．図 4 に，重力計算の精度を制御するパラメー

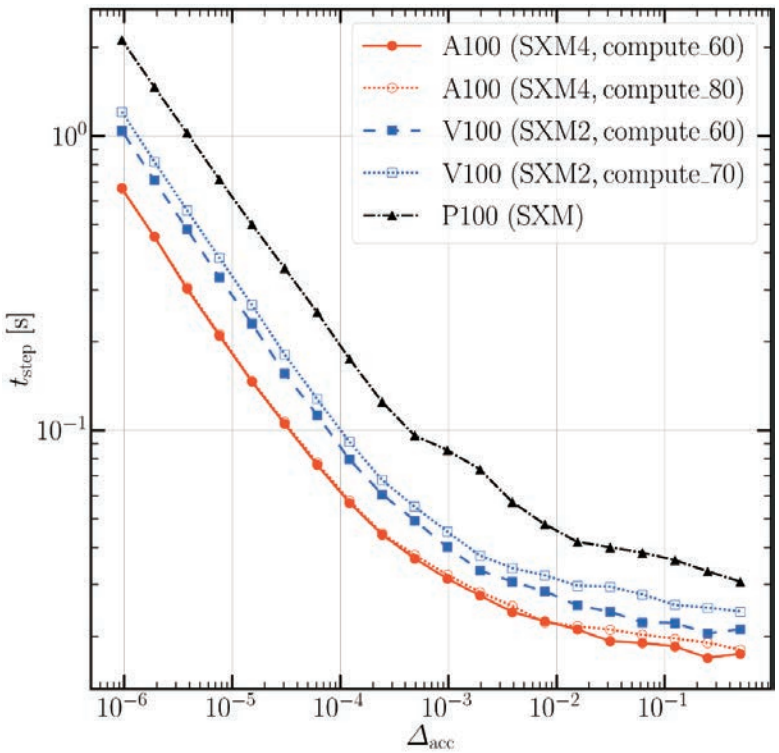


図 4　GOTHIC の実行時間．重力計算の精度を制御するパラメータ$\Delta_{\rm acc}$の関数として，ステップ当たりの実行時間を示した．

タΔ_{acc}の関数としてステップあたりの実行時間を示した．パラメータΔ_{acc}の値が小さいほど重力計算は正確であるので，測定結果が左下方向へ移動するほど高性能であるということを意味する．図 4 からは，赤で示した A100 の測定結果が常に P100 や V100 よりも高速であることが分かる．また，V100 および A100 においては Independent Thread Scheduling の有効時（open symbols）と無効化時（filled symbols）の結果も示した．V100 においては Independent Thread Scheduling を無効化することで 10-20%の高速化が達成されていたが，A100 においては依然無効化時の方が速いものの両者の差は縮まっている．これには，warp-wide reduction による高速化が Independent Thread Scheduling 有効時にしか得られないことが寄与していると考えられる．

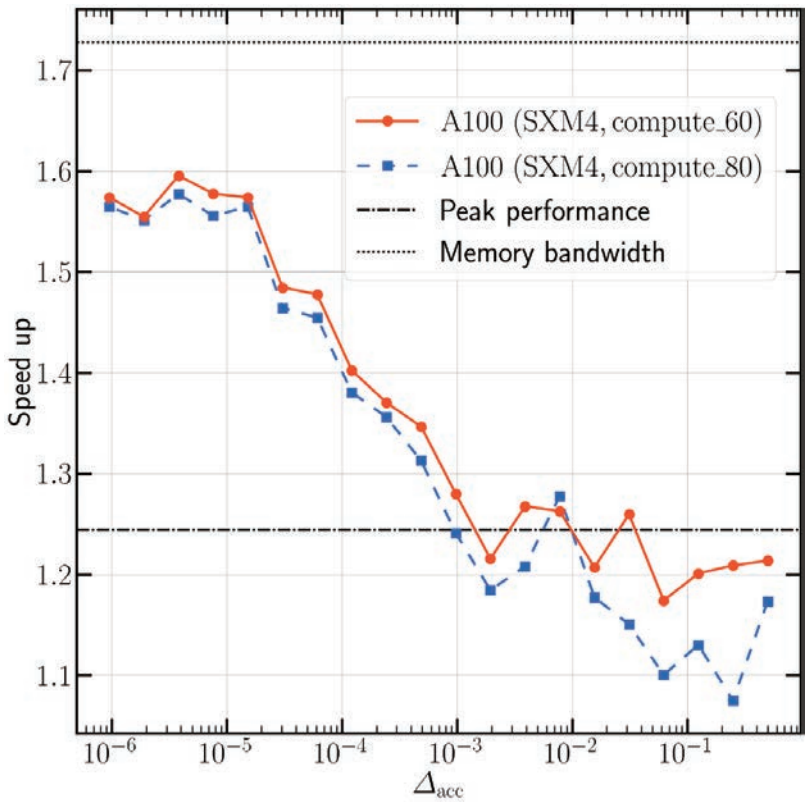


図 5　GOTHIC の速度向上率．

図 5 には，V100 からの速度向上率を示す．一点鎖線は V100 と A100 の単精度理論ピーク性能比，点線はメモリバンド幅比を示している．得られた速度向上率は Independent Thread Scheduling の有無にはあまり依らずにおおむね理論ピーク性能比よりも高い値で推移し，最大で 1.6 倍に達することが分かった．理論ピーク性能比よりも高い速度向上率が得られた理由は分かっていないが，シェアードメモリの容量増加や今回採用したパラメータセットの特性といった要因が考えられる．前者は SM あたりに割り当てられるブロック数の増加につながり，その結果として浮動小数点演算，整数演算，グローバルメモリへのアクセスといった命令の同時実行が容易となり，実行時間の隠ぺいがなされるということである．また後者については，性能に関係するパラメータの設定を行ったマイクロベンチマークは典型的な精度である$\Delta_{acc} = 2^{-9}$を仮定していたため，他の値においては最適な構成とは異なるパラメータセットを使用している可能性があ

る．これは A100・V100 に共通しているため，理論ピーク性能比よりも高い速度向上率の原因となっているかもしれない．

5. まとめ

Wisteria/BDEC-01（Aquarius）に搭載された A100 を使いこなすために，本稿では A100 および CUDA 11 において導入された新機能とその使い方を紹介した．また，活用事例として重力ツリーコード GOTHIC の性能評価結果を紹介した．GOTHIC は一世代前の GPU である V100 上での性能測定結果と比べて最大で 1.6 倍高速であり，演算律速な問題においても理論ピーク性能比よりも高い速度向上率を示す実例があることを示した．

参 考 文 献

[1] Wisteria/BDEC-01: https://www.cc.u-tokyo.ac.jp/supercomputer/wisteria/system.php

[2] NVIDIA Corporation: NVIDIA Tesla P100 (2016)

[3] NVIDIA Corporation: NVIDIA Tesla V100 GPU Architecture (2017)

[4] NVIDIA Corporation: NVIDIA A100 Tensor Core GPU Architecture (2020)

[5] Miki, Y.: Gravitational Octree Code Performance Evaluation on Volta GPU, Proceedings of the 48th International Conference on Parallel Processing, ICPP 2019, New York, NY, USA, ACM, pp. 62:1-62:10 (2019)

[6] NVIDIA Corporation: CUDA C++ Programming Guide Version 11.5 (2021)

[7] Svedin, M., Chien, S. W. D., Chikafa, G., Jansson, N. and Podobas, A.: Benchmarking the Nvidia GPU Lineage: From Early K80 to Modern A100 with Asynchronous Memory Transfers, HEART '21: Proceedings of the 11th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies, 9:1-9:6 (2021)

[8] 三木洋平： NVIDIA A100 における重力ツリーコードの性能評価，研究報告ハイパフォーマンスコンピューティング（HPC），Vol. 2021-HPC-180, No. 24, pp. 1-7 (2021)

[9] Miki, Y. and Umemura, M.: GOTHIC: Gravitational oct-tree code accelerated by hierarchical time step controlling, New Astronomy, Vol. 52, pp. 65-81 (2017)

[10] Miki, Y. and Umemura, M.: MAGI: many-component galaxy initializer, Monthly Notices of the Royal Astronomical Society, Vol. 475, pp. 2269-2281 (2018)

Numerical simulation of deep-water oil blowouts

Daniel Cardoso Cordeiro

大阪大学基礎工学研究科

1. Introduction

In 2010, the largest offshore blowout of all time happened in the Gulf of Mexico, USA. The accident became known as the Deep-water Horizon oil spill in which around 4.9 million oil barrels were spilled into the ocean, and chemical dispersants were applied in a large scale into the subsea as a way to treat the oil blowout plume [1].

Given the possible negative outcomes that subsea injection of chemical dispersants may cause to the environment, it is necessary to assess clearly, whether this technique is specifically adequate for a certain accident scenario before its application. However, since real-scale experiments are unpractical due to their obvious environmental risk, numerical models and laboratory-scale experiments are the only available tools to investigate the blowout phenomenon. Therefore, the development of an accurate numerical blowout model would represent a valuable asset in the decision-making process for the remediation of such accidents with chemical dispersants.

A number of challenges stand in the way of achieving a more exact blowout model. Among them are the sensitivity of model predictions to complex chemical and biological phenomena, the need for a coupling of ocean circulation models with the near-field blowout plume dynamics and the accuracy of the oil/gas droplet size distribution (DSD) [2]. Particularly, the droplet size distribution (DSD) represents a serious threat to the preciseness of the spilled oil path inside the sea as a small variation in the droplet diameter results in a large discrepancy in the distance that the oil could travel both laterally and vertically inside the ocean. This could induce the response team to choose unsuitable remediation techniques with further damage to the environment.

In this research, we propose the use of a LES (Large Eddy Simulation) for turbulence with a hybrid Euler-Euler and volume of fluid (VOF) models to accurately predict the DSD by calculating the actual breakup and coalescence caused by the atomization of the turbulent oil jet into a quiescent water media.

2. Numerical methods

To simulate the oil blowout phenomena, a hybrid Eulerian multifluid solution framework with a volume of fluid (VOF) sharp interface capturing algorithm was used. It improves the accuracy of the traditional Euler-Euler method without the computational burden of coupling an entire interface-capturing model [3]. The present model was validated against a series of laboratory-scale oil blowout experiments. The deterministic approach based on the large-eddy simulation does not rely on the input of any empirical

data nor calibration parameters, increasing the reliability to the real-scale prediction of the DSD in deep-water blowouts.

The governing equations solved are the Continuity (Eq. 1) and Navier-Stokes (Eq. 2) equations:

$$\frac{\partial \alpha_k}{\partial t} + \boldsymbol{u_k}.\nabla\alpha_k = 0 \tag{1}$$

$$\frac{\partial(\rho_k\alpha_k\boldsymbol{u_k})}{\partial t} + (\rho_k\alpha_k\boldsymbol{u_k}.\nabla)\boldsymbol{u_k} = -\alpha_k\nabla p + \nabla.(\mu\alpha\nabla\boldsymbol{u_k}) + \rho_k\alpha_k\boldsymbol{g} + \boldsymbol{F_{D,k}} + \boldsymbol{F_{S,k}} + \boldsymbol{F_{vm,k}} \tag{2}$$

where u is the velocity, is the volume fraction, t is time, ρ is the density, p is the pressure, g is the gravity acceleration, $F_{S,k}$ is the surface tension force, $F_{D,k}$ is the drag force, $F_{vm,k}$ is the virtual mass force and the subscript k indicates the fluid phase. The Euler-Euler method was used to model the multiphase flow coupled with a Schiller-Naumann drag model. The open-source OpenFOAM v4.0 was used for the calculations.

Table 1: Configuration of the cases: dispersants to oil ratio, DOR; surface tension, σ [mN/m]; inlet diameter, D [mm], inlet velocity, U_{in} [m/s]; the jet Reynolds number, Re; the jet Weber number, We; the oil-water viscosity ratio, ν_o/ν_w; the oil-water density ratio, ρ_o/ρ_w ; simulated time interval, ΔT.

	Case	DOR (%)	σ	D	U_{in}	Re	We	ν_o/ν_w	ρ_o/ρ_w	$\Delta TU_{in}/D$
——	1	0%	28.3	14	0.65	1076	179	8.46	0.856	428
– – – –	2	1%	0.6	14	0.65	1076	8438	8.46	0.856	428
·····	3	0%	28.3	9	0.65	691	115	8.46	0.856	667
—▲—	4	0%	28.3	14	0.16	264	11	8.46	0.856	114
– –△– –	5	1%	0.6	14	0.16	264	511	8.46	0.856	114
—●—	6	0%	28.3	14	0.65	2151	179	4.23	0.856	428
– –○– –	7	0%	28.3	14	0.65	4302	179	2.11	0.856	428
—■—	8	0%	28.3	14	0.65	1076	125	8.46	0.600	428
– –□– –	9	0%	28.3	14	0.65	1076	84	8.46	0.400	428

The domain was modeled based on the experimental data [4], as a 1.0 m height cuboid tank with a 0.3 m per 0.3 m width. Oil flows through a round inlet at the center of the tank initially filled with water. The DSD was computed a posteroiri by isolating the droplet in the isosurface alpha = 0.1.

3. Results and Discussion

The effects of different parameters over the jet atomization in the DSD were showed using non-dimensional numbers and figures of the jet behavior.

The velocity inlet altered the regime in different cases: 1-2 were atomization and 4-5 were second wind induced breakup. Both the regimes showed very different characteristics regarding the droplet size distribution and the velocity of the particles as per Fig. 1.

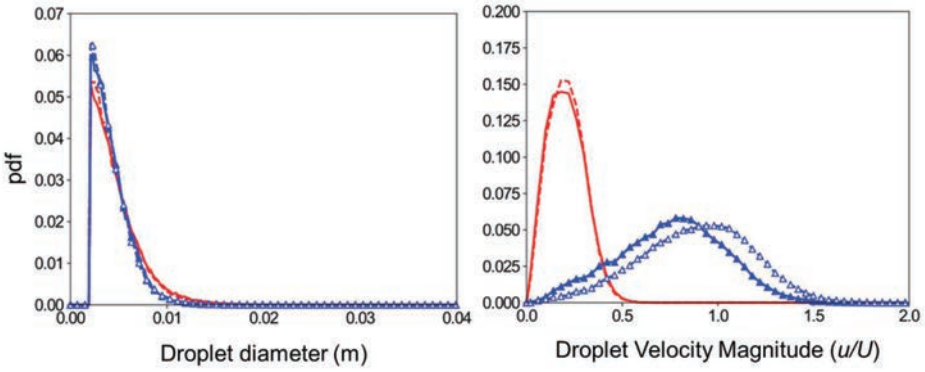


Figure 1: Droplet diameter distribution (left) and droplet velocity distribution (right) for cases 1, 2, 4 and 5.

From the droplets local information, it was possible to conclude that for the case of surface tension reduction, the droplet diameter was slightly reduced while the droplet velocity distribution behaved differently for the cases in different breakup regimes. As Fig. 1 also shows, in the atomization regime (Cases 1 and 2), the droplet velocity was reduced and increased in the second wind induced regime (Cases 4 and 5).

The inlet diameter reduction did not change the flow regime; however, it intensified Kelvin-Helmholtz instabilities near the inlet and narrowed the droplet size distribution (Fig. 2).

Figure 2: Effect of the diameter inlet reduction in the snapshots of the isosurface alpha = 0.1 for cases 1 (left) and 3 (right). The details show the Kelvin-Helmholtz instabilities.

The cases with variation in the density (1, 8) and the viscosity ratio (1, 6), showed in Figs. 3 and 4 respectively, illustrates the impact of the near-field instabilities in both the droplet size distribution and in the droplet velocity distribution. The viscosity reduction decreased the Kelvin-Helmholtz effect while keeping similar DSD, which is caused by the increase in the droplet Re. The density reduction acted to enhance the instabilities near the inlet, inducing breakup much

earlier than in the higher density cases.

Figure 3: Effect of the viscosity ratio reduction in the snapshots of the isosurface alpha = 0.1 for cases 1 (left) and 6 (right). The details show the Kelvin-Helmholtz instabilities near the inlet.

Figure 4: Effect of the density ratio reduction in the snapshots of the isosurface alpha = 0.1 for cases 1 (left) and 8 (right). The details show the Kelvin-Helmholtz instabilities near the inlet.

4. Conclusion

Using the hybrid volume-of-fluid/Euler-Euler model with a large eddy simulation turbulent model was effective in modeling the laboratory-scale oil blowout in a water tank. The effects of different physical parameters on the atomization were complex and it shows the difficulty in understanding how the distribution of droplets are affected by this phenomenon.

Further research is necessary to create and overarching theory that could explain the jet breakup in a more clear and concise way.

References

[1] Kujawinski et al., Environ. Sci. Technol., vol. 45, pp. 1298-1306, 2011.

[2] Socolofsky et al., Oceanography, vol. 29, pp. 64-75, 2016.

[3] Wardle, Weller, International Journal of Chemical Engineering, vol. 13, 2013.

[4] Miyata et al., JIME, vol. 51, pp.367-374, 2016.

Vlasov 方程式に基づく金属の光吸収シミュレーション

谷　水城

東京大学大学院工学系研究科原子力国際専攻

１．背景と概要

フェムト秒レーザーアブレーションは高強度光から電子へのエネルギー伝達に始まり、電子系のエネルギーが格子系へと次第に移行することで爆発が生じた結果、固体材料の表面に窪みを生じさせるテクニックである。ピコ秒以上に長いパルスレーザーを用いた場合には、加工痕近傍に熱の影響で変質が生じるが、フェムト秒レーザーではこれが抑制されることが知られている。しかし、材料に所望の加工を施すためには、職人の勘と経験に基づいてレーザー照射の条件を色々変えてテスト加工と計測を繰り返し行うことが必要とされ、数ヶ月から数年かかることも珍しくない。そこで、レーザー加工の学理を構築し、それに基づいてコンピュータシミュレーションによりサイバー空間でのテスト加工を高速に行うことが可能になれば、レーザー照射条件出しにかかる時間を大幅に短縮できることが見込まれる。そのためには、フェムト秒レーザー照射によるエネルギーの時空間ダイナミクスを明らかにすることが必要である。

ところが、光-電子エネルギー移行から加工痕が生じるまでに生じる過程は時間スケール・空間スケール共にマルチスケールであるため、多階層にまたがったモデリングが必要となる。まず我々は金属のフェムト秒加工の最初期過程である電子の光エネルギー吸収のモデリング方法を検討する。これにはプラズマモデルや連続体モデルを用いて記述することが一般的だが、温度が定義できない非平衡状態を記述することはできない。非平衡状態を記述するには時間依存密度汎関数理論(TDDFT)がよく用いられるが、多大な計算コストを必要とするため、現代のスーパーコンピュータを用いたとしてもレーザーパラメータ最適化に用いるのは困難である。そこで注目したいのが、原子核衝突や金属原子クラスターの光応答を計算するのに用いられる Vlasov-LDA 方程式[1]である。本研究では、Vlasov-LDA 方程式を固体金属にも適用できるように、周期境界条件を課して計算コストの低い擬似粒子法を用いて解く手法を開発し、アルミニウムの電子の光エネルギー吸収量の評価では、TDDFT の 1/50 程度の計算時間で、TDDFT とよく一致する結果が得られることを明らかにした[2]。

２．定式化

時間依存密度汎関数理論の基礎方程式は時間依存 Kohn-Sham(KS)方程式

$$i\hbar\frac{\partial}{\partial t}\psi_n(r,t) = \mathcal{H}(r,t)\psi_n(r,t)$$

$$\mathcal{H}(r,t) = \frac{p^2}{2m} + V_{\mathrm{eff}}[n_e](r,t)$$

である。ここに、ψ_nは KS 軌道、V_{eff}は電子密度n_eに依存する有効ポテンシャルで、原子核と電子の Coulomb ポテンシャルと電子間の Hartree ポテンシャルと Perdew-Zunger の局所密度近似(LDA)交換相関ポテンシャル[3]と光電場の作るポテンシャル(電気双極子近似、長さゲージ)よりなる。電子密度n_eは

$$n_e(r,t) = \sum_n \psi_n(r,t)\psi_n^*(r,t)$$

密度演算子は

$$\rho(r,r',t) = \sum_n \psi_n(r',t)\psi_n^*(r,t)$$

により定義される。この時間発展は von-Neumann 方程式

$$\frac{\partial}{\partial t}\rho(t) = -\frac{i}{\hbar}[\mathcal{H}(t),\rho(t)]$$

により記述することができて、Wigner 変換を施せば$\rho(t)$は位相空間上の一体分布関数$f(r,p,t)$へと写像されて、$\hbar \to 0$の極限をとることにより$f(r,p,t)$の時間発展を記述する Vlasov-LDA 方程式

$$\frac{\partial}{\partial t}f(r,p,t) = -\frac{p}{m}\cdot\nabla_r f(r,p,t) + \nabla_r V_{\mathrm{eff}}[n_e](r,t)\cdot\nabla_p f(r,p,t)$$

が得られる。電子密度n_eと分布関数$f(r,p,t)$は

$$n_e(r,t) = \int f(r,p,t)d^3p$$

により結びつく。Vlasov-LDA 方程式で用いる有効ポテンシャルV_{eff}では、原子核と電子の間のクーロンポテンシャルの代わりに、Heine-Abarenkov 型の局所擬ポテンシャル[4]を用いることで価電子のみを計算の対象とし、計算コストの削減を図る。各時刻での電流密度は

$$J(t) = \frac{1}{|\Omega|}\iint_\Omega \left(-e\frac{p}{m}\right) f(r,p,t)d^3r d^3p$$

で計算できる。ここに、$|\Omega|$は直方体の形をしたシミュレーションボックスΩの体積である。

３．擬似粒子法

Vlasov-LDA 方程式を数値的に解くために、電子密度n_eとその汎関数は空間グリッドにより離散化する。ただし、$f(r,p,t)$を直接６次元空間のグリッド上で扱うと計算が大変なので、こちらは擬似粒子法を用いる。擬似粒子法では、N_e電子系の$f(r,p,t)$はN_{pp}個の擬似粒子の集合で近似し

$$f(r,p,t) = \frac{N_e}{N_{pp}}\sum_i^{N_{pp}} g_r[r - r_i(t)]g_p[p - p_i(t)]$$

とする。典型的には１電子あたり１万個のオーダーの擬似粒子を用意する必要がある。g_r, g_pは擬似粒子が位相空間の中で持っている重み分布で、それぞれi番目の粒子の中心が$r_i(t), p_i(t)$となっていて、形状は標準偏差が$\sqrt{2}\sigma_r, \sqrt{2}\sigma_p$のガウス関数である。電子密度と電流密度は

$$n_e = \frac{N_e}{N_{pp}}\sum_i^{N_{pp}} g_r[r - r_i(t)]$$

$$J(t) = -\frac{1}{|\Omega|}\frac{N_e}{N_{pp}}\sum_i^{N_{pp}} e\frac{p_i(t)}{m}$$

で評価することができる。また、Vlasov-LDA 方程式自体は各擬似粒子についての Newton 方程式へと帰着され、

$$\dot{r}_i = \frac{p_i}{m}, \dot{p}_i = -\int_\Omega V_{\text{eff}}(r)\nabla_{r_i} g_r(r_i - r)\, d^3r$$

を解けばいいということになる。数値積分は Verlet 法を用い、並列化は MPI を用いたプロセス並列のみで、各プロセスに均等に粒子とグリッド点を振り分ける。周期境界条件を課すので、シミュレーションボックスの外に擬似粒子が出てしまった場合は、反対側から戻ってくるようにし、Hartree ポテンシャルと原子核-電子間のポテンシャルは高速 Fourier 変換(FFT)を使って計算する。

４．初期状態の準備

初期状態は Vlasov-LDA 方程式の定常解を与える、化学ポテンシャルをμ とした Thomas-Fermi(TF)方程式

$$\frac{\hbar^2}{2m}[3\pi^2 n_e(r)]^{2/3} + V_{\text{eff}}(r) = \mu$$

により求める。その手続きを第１図に示す。一旦$n_e(r)$を与えると$V_{\text{eff}}(r)$が決まるので、TF 方程式に代入すると、全電子数N_eになるように新しいμと$n_e(r)$を決めることができて、これを繰り返すことで、自己無撞着に解に到達できる。

```
1:  procedure GROUND STATE PREPARATION
2:      (–Initialization–)
3:      initial guess of μ and n_e^in
4:
5:      (–Self-consistent determination of μ and n_e–)
6:      while Δn > ε (ε = 10^−7) do
7:          set pseudo-particle position {r_i}
8:          {r_i} ↦ n_ps(r) using Eq. (25)
9:          n_ps ↦ V_eff[n_ps](r) using Eq. (33)
10:         V_eff[n_ps](r) ↦ n_e
11:         find appropriate μ
12:         Δn = ∫_Ω dr|n_e^in(r) − n_e(r)|
13:         n_e^in = n_e
14:     end while
15:
16:     (–Set Pseudo-particle Momenta–)
17:     for i = 1, N_pp do
18:         while p > p_f do
19:             random number p_x (0 ≤ |p_x| ≤ p_f)
20:             random number p_y (0 ≤ |p_y| ≤ p_f)
21:             random number p_z (0 ≤ |p_z| ≤ p_f)
22:             p = √(p_x^2 + p_y^2 + p_z^2)
23:         end while
24:         p_i = (p_x, p_y, p_z)
25:     end for
26: end procedure
```

第 1 図：TF 方程式を解く部分の実装

While 文のところで自己無撞着に化学ポテンシャルμと電子密度n_eを求める。その次の for 文で、先ほど求めた電子密度を再現するように擬似粒子を配置させる。

十分に$n_e(r)$が収束したら、求めた$n_e(r)$を再現するように擬似粒子を配置する必要がある。その手続きを第２図に示す。$n_e(r)$はグリッド上で値を持っているので、そのグリッド近傍に擬似粒子を一様に分布するように擬似乱数で位置をばらつかせながら注意深く配置する。

procedure HOW TO SET PSEUDO PARTICLE POSITION
(–# of Pseudo Particles around $\mathbf{r}_s$–)
a sub-grid point $\mathbf{r}_s = (x_s, y_s, z_s)$
calculate $n_e^{\rm inp}(\mathbf{r}_s)$ by interpolation
the number of pseudo particles $n_e^{\rm inp} \mapsto N_{\rm pp}^{\rm local}$

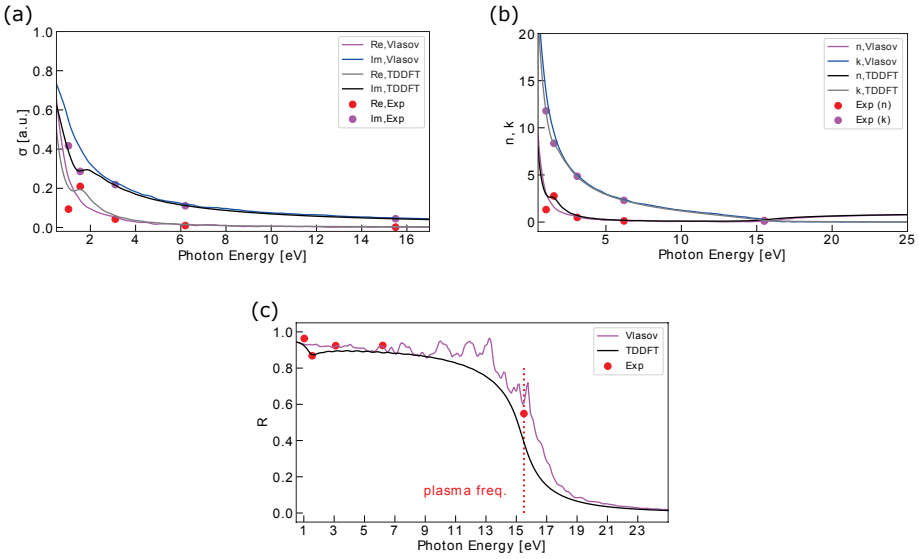
(–Distribute Pseudo Particles around $\mathbf{r}_s$–)
for $l = 1, N_{\rm pp}^{\rm local}$ **do**
random number R_x $(-0.5 \le R_x \le 0.5)$
random number R_y $(-0.5 \le R_y \le 0.5)$
random number R_z $(-0.5 \le R_z \le 0.5)$
$x_l = x_s + R_x L_x / N_{\rm inp}^x$
$y_l = y_s + R_y L_y / N_{\rm inp}^y$
$z_l = z_s + R_z L_z / N_{\rm inp}^z$
l-th pseudo particle position is
$\mathbf{r}_l = (x_l, y_l, z_l)$
end for
end procedure

第２図：擬似粒子の配置手続き

各グリッド上の局所電子密度から、何個の擬似粒子を置けばいいのかをまず計算する。次に、グリッド点を中心とする立方体(一辺はグリッド幅に等しい)中に擬似乱数を用いて一様に配置する。

５．線形光学応答

計算の対象はバルクアルミニウムで、立方体の面心立方格子をシミュレーションボックスに取る。Vlasov-LDA のグリッド幅は 0.5025Å、擬似粒子の数は 10000 粒子/電子、幅はσ_r = 0.5779 Å、電子数は 12 個、時間刻み幅は 0.0060475 as である。σ_rは以下に見る光学応答が 2 eV-25 eV の領域で TDDFT と一致するように選んだ。MPI のプロセス数は 64 プロセスとする。TDDFT はオープンソースソフトウェアの SALMON[5]を用いて計算し、グリッド幅は 0.2871Å、k 空間は 48^3 個サンプリングし、時間刻み幅は 0.036285 as である。MPI のプロセス数は 7168 プロセスとする。いずれの手法でも時刻 t=0 で全ての電子に小さな運動量を与えて、インパルス応答から光学スペクトルを計算することができる。線形光学応答の計算結果を第３図に示す。



第３図：線形応答

(a)光学伝導度：ピンクと青の実線が Vlasov-LDA で計算した実部と虚部、灰色と黒の実線が TDDFT で計算した実部と虚部、赤とピンクのドットが実験値[6]から参照した実部と虚部の値である。(b)複素屈折率：ピンクと青の実線が Vlasov-LDA で計算した屈折率と消衰係数、灰色と黒の実線が TDDFT で計算した屈折率と消衰係数、赤

とピンクのドットが実験値[6]から参照した屈折率と消衰係数の値である。(c)反射率：ピンクの実線が Vlasov-LDA で計算したもの、黒の実線が TDDFT で計算したもの、赤のドットが実験値[6]から参照したものである。実験値は全て 1.03 eV (1200 nm), 1.2 eV (1030 nm), 1.55 eV (800 nm), 3.1 eV (400 nm), 6.2 eV (200 nm), and 15.5 eV (80 nm)の５点取ってきている。

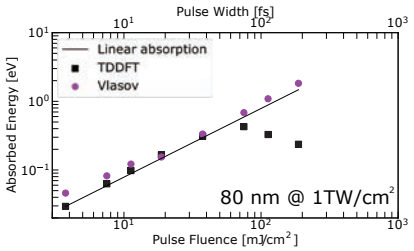
いずれも Vlasov-LDA と TDDFT と実験値が概ね一致している。1.5 eV 付近では TDDFT の結果と実験値にバンド間遷移に由来する異常分散が見られるのに対して、Vlasov-LDA では正常分散となっている。したがって、Vlasov-LDA ではバンド間遷移は適切に捉えられないが、それ以外の光子エネルギー領域ではよく TDDFT と実験値を再現できている。

６．光-電子エネルギー移行シミュレーション

レーザー電場は

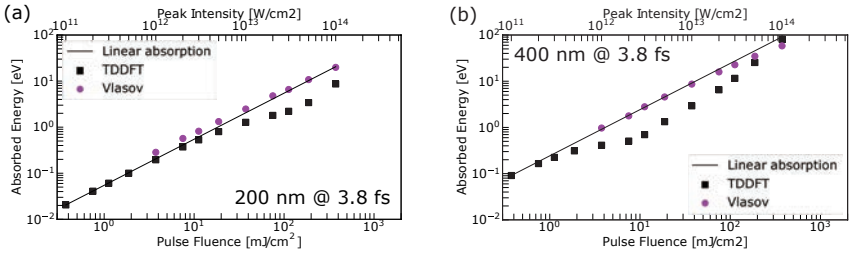
$$E(t) = E_0 \sin\left[\hbar\omega\left(t - \frac{T}{2}\right)\right] \sin^2\left(\frac{t\pi}{T}\right) \ (0 \le t \le T)$$

という時間プロファイルを採用する。ここに、E_0は電場振幅、$\hbar\omega$は中心光子エネルギー、Tはパルスの立ち上がりから立ち下がりまでの時間幅である。なお、パルス幅は強度の半値全幅(FWHM)では 0.36T 程度となる。エネルギーの吸収量は Vlasov-LDA でも TDDFT でも全エネルギーの増分によって評価する。すなわち、レーザーパルス照射前後のエネルギーの増分を吸収したエネルギーとする。第４図に中心波長 80 nm、ピーク強度10^{12} W/cm^2 の場合のエネルギー吸収量のパルス幅依存性を、第５図に中心波長 200 nm および 400 nm、パルス幅(FWHM)3.8 fs の場合のエネルギーの吸収量の光強度依存性を示す。



第４図：エネルギー吸収量のパルス幅依存性

中心波長 80 nm、ピーク強度 10^{12} W/cm^2 の場合について、ピンクの点が Vlasov-LDA の結果、黒の点が TDDFT の結果を示しており、黒の実戦は TDDFT の低フルエンス の領域から線形吸収を仮定して外挿した補助線である。



第５図：エネルギー吸収量のパルス強度依存性

(a)が中心波長 200 nm、(b)が中心波長 400 nm で、いずれもパルス幅(FWHM)3.8 fs の場合について、ピンクの点が Vlasov-LDA、黒の点が TDDFT の結果を示しており、黒の実線は TDDFT の低フルエンス領域から線形吸収を

仮定して外挿した補助線である。

まず中心波長 80 nm の場合、パルス幅 100 fs 以上では TDDFT の場合に線形吸収ではなくなるのに対して、Vlasov-LDA では線形吸収になっていることが見て取れる。TDDFT の場合はコヒーレントなバンド間遷移の振動によって長いパルスの場合に Rabi 振動のような現象が生じることが知られていて、今回の結果もこれに該当するものと考えられる。対照的に、線形応答でみたように、Vlasov-LDA ではバンド間遷移が入らないため長いパルスでも線形吸収となっていると理解できる。中心波長 200 nm および 400 nm の場合、TDDFT の場合に線形吸収ではなくなる強度領域があるのに対して、Vlasov-LDA では線形吸収になっていることが見て取れる。強度を上げていくと吸収量が回復することも併せて考えると、TDDFT で多くの電子が励起してしまったために吸収飽和が生じていると考えられる。Vlasov-LDA については線形応答やパルス幅依存性で見たようにバンド間遷移が入らないため吸収飽和が生じず、全ての強度領域にわたって線形吸収となっているものと理解できる。これらの光吸収シミュレーションは一点計算するのに 1 プロセス換算で TDDFT は 975 時間なのに対して、Vlasov-LDA では 24 時間であった。これは Vlasov-LDA が TDDFT の約 50 倍効率的に計算できるということを意味する。

7．さいごに

TDDFT とその半古典極限としての Vlasov-LDA 方程式を導入し、周期境界条件を課すことでバルクアルミニウムの線形応答と光エネルギー吸収を評価した。その結果、Vlasov-LDA ではバンド間遷移こそ入らないものの、TDDFT から予想される線形吸収のトレンドを非常によく再現することがわかった。Vlasov-LDA の計算コストは TDDFT の約 1/50 であり、より大きな系の計算に対して有望であることを示唆している。また、非周期系の Vlasov-LDA で、TDDFT では難しい電子間散乱を考慮する方法が確立されているので、周期系に対しても同様に電子間散乱による緩和の効果を調べることができると予想される。電子間散乱は高温で重要な緩和過程なので、これを計算できる可能性はレーザー加工シミュレーションの観点から望ましいものである。今後はこの電子間散乱に加えて、電磁場解析も同時に行うことで、より大きなシステムに対するマルチスケールシミュレーションの手法を確立して行きたいと考えている。

謝辞

本研究は東京大学情報基盤センター 2020 年度若手・女性利用者推薦制度採択課題「半古典輸送理論に基づいた電子・光融合シミュレーション」の助成を受けて実施された。

参 考 文 献

1) G. F. Bertsch and S. D. Gupta, Phys. Rep. **160**, 189 (1988).
2) M. Tani, T. Otobe, Y. Shinohara, and K. L. Ishikawa, Phys. Rev. B **104**, 075157 (2021).
3) J. P. Perdew and A. Zunger, Phys. Rev. B **23**, 5048 (1981).
4) C. Fiolhais, J. P. Perdew, S. Q. Armster, J. M. MacLaren, and M. Brajczewska, Phys. Rev. B **51**, 14001 (1995).
5) M. Noda, *et al.*, Comput. Phys. Commun. **235**, 356 (2019).
6) A. D. Rakic, Appl. Opt. **34**, 4755 (1995).

散開星団起源ブラックホール連星の形成と特徴

熊 本 淳

東京大学理学系研究科

１．はじめに

近年、LIGO と Virgo による重力波イベントの検出が続いている[1]。これまでに検出された連星ブラックホールの合体イベントにより、近傍宇宙における連星ブラックホールの合体率や質量分布等が推定されている。さらに、連星ブラックホールの観測から得られるパラメータの一つに有効スピンパラメータ $\chi_{\rm eff}$ がある。連星ブラックホールの有効スピンパラメータの分布についても推定が行われており、$\chi_{\rm eff}\sim 0$ 付近に小さな正負の値を持って分布していることが予想されている。このようなスピン分布の理解が連星ブラックホールの起源を考える上で重要な要素の一つとなる。

我々はこれまでの研究において、散開星団における連星ブラック―ル形成について研究を行って来た。金属量が異なる散開星団について、重力 N 体シミュレーションコード NBODY6++GPU[2]を用いて計算を行い、散開星団内における連星ブラックホールの形成過程について調べてきた。今回は、4 つの異なる金属度を持つ散開星団について重力 N 体シミュレーションを行い、それぞれの星団で形成される連星ブラックホールの前駆体について、金属量に依存した恒星の質量損失・スピン損失を仮定し、スピン進化を計算した。

２．シミュレーションの方法と星団モデル

本研究では数千個の星からなる星団を重力多体系として扱うシミュレーションを行う。本研究で扱う星団は高密度であり，それぞれの星の進化に伴う半径の変化や質量放出等の非線形な物理過程が複雑に絡み合うため，多くの計算コストが必要となる。また，連星進化と星団全体の進化ではタイムスケールが大きく異なるため，これらの過程を同時に計算するために，大規模な計算機を用いた数値シミュレーションによる研究が必要不可欠となる。そこで，重力 N 体シミュレーションコード NBODY6++GPU[2]を用いて，Reedbush-L にて計算を行った。

NBODY6++GPU は星団計算用の重力多体計算コード NBODY6 を並列計算機，GPU 計算機対応にアップデートしたもので，より高速に計算ができる。個々の星の運動については４次のエルミート法を用いて計算する。今回のシミュレーションでは，初期に連星は存在しないが星の重力三体相互作用により連星を形成する。これらの連星の運動を計算するためのタイムステップは星団全体の計算のためのタイムステップよりもかなり短くなってしまう。そこで，連星の軌道計算のために KS 法と呼ばれる方法が本シミュレーションコードには実装されている。

２．１．初期条件

本研究で扱う星団の初期条件として，Plummer model を採用した。このモデルの密度分布ρ(r)は以下の式で与えられる。

$$\rho(r) \propto \left(1+\frac{r^2}{r_{\rm p}^2}\right)^{-5/2}$$

ここで，rおよびr_pは星団中心からの半径およびスケール半径である。本研究ではスケール半径は，0.31pc とした。また，星団を構成する各星の初期質量は 0.08 から 150 太陽質量として，質量分布は Kroupa の初期質量関数[3]に従うように乱数で与えた。

表１に今回用いた星団モデルの初期条件について，主なパラメータをまとめる。2500 太陽質量の星団について金属量の異なる４つのモデルについて計算を行った。

表１：シミュレーションモデル

	星団質量 ［太陽質量］	金属量 ［太陽金属量］	ラン数
Model A	2500	0.1	360
Model B	2500	0.025	500
Model C	2500	0.5	1000
Model D	2500	1	1000

２．２．星の進化

今回用いたシミュレーションコードには星の進化モデルが実装されている。このモデルでは，初期質量，金属量の関数として恒星の質量や半径の時間進化を計算する。Z＝0.02 が太陽金属量に相当する。金属量が大きいモデルでは恒星が進化する時に星風によって多くの質量が失われる。そのため，最終的に形成されるブラックホールの質量は小さくなる。

類似の先行研究では球状星団に着目したものが大多数である(例：[4]，[5])。散開星団に着目した先行研究も存在するが初期から連星を加え，その後の連星進化を調べている(例：[6])。我々の研究の特色は散開星団に着目し，連星形成過程から調べた点である。

３．スピン進化

N体シミュレーションの結果形成された連星ブラックホールについてスピン進化の計算を行う。本研究ではスピンを形成する要因として common envelope 後の Wolf-Rayet 星が伴星からの潮汐力によってスピンアップする過程を考える[7]。この過程の概念図を図１に示す。

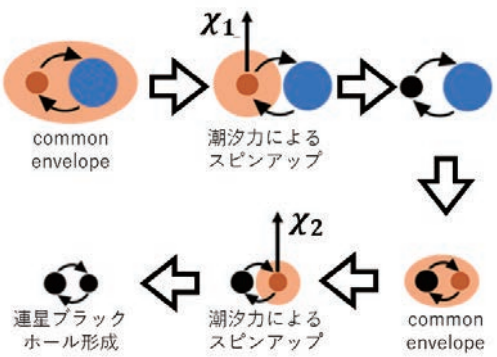


図１：連星進化過程における星のスピンアップの概念図

伴星からの潮汐力によるスピンアップによって得られる各星のスピンχ_1, χ_2は以下の式によっ

て計算する。

$$\frac{d\chi_{1,2}}{dt} = \frac{\chi_{syn} - \chi_{1,2}}{t_{syn}} - \frac{\chi_{1,2}}{t_w}$$

ここで、t_{syn}, t_wはそれぞれスピンアップのタイムスケールと質量損失のタイムスケールである。また、完全にスピンアップした時のスピンパラメータがχ_{syn}である。これらのパラメータは以下の式で与えられる。

$$t_{syn} \sim 10q^{-\frac{1}{8}}\left(\frac{1+q}{2q}\right)^{\frac{31}{24}}\left(\frac{t_c}{1\mathrm{Gyr}}\right)^{\frac{17}{8}}\ \mathrm{Myr}$$

$$\chi_{syn} \sim 0.5q^{\frac{1}{4}}\left(\frac{1+q}{2}\right)^{\frac{1}{8}}\left(\frac{\epsilon}{0.075}\right)\left(\frac{R_{2,1}}{R_{sun}}\right)^{2}\left(\frac{m_{2,1}}{30M_{sun}}\right)^{-\frac{13}{8}}\left(\frac{t_c}{1\mathrm{Gyr}}\right)^{-\frac{3}{8}}$$

$$t_w \propto Z^{-0.86}$$

ここで、q, t_cは連星を構成する星の質量比と連星の合体時間であり、R_*, m_*は連星を構成する各星の半径と質量である。

4．結果

シミュレーションで形成された連星について、スピンパラメータを計算した結果、軌道長半径の小さい連星は、伴星からの強い潮汐力によって、より大きな有効スピンを持つ連星ブラックホールに進化することがわかった。図 2 は各金属量のモデルについて得られた連星ブラックホールの有効スピンの累積規分布を示す。

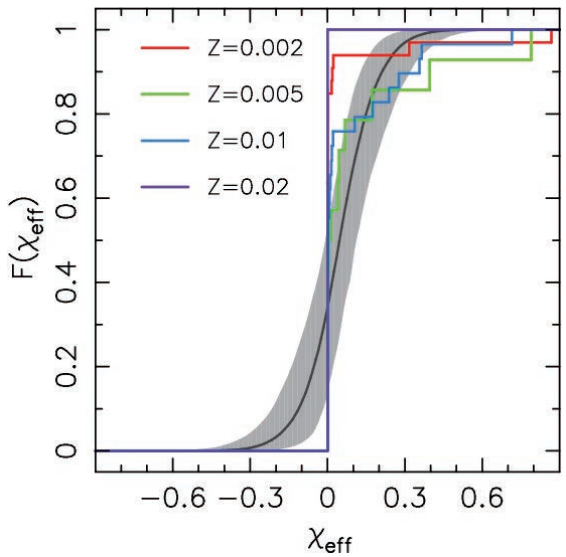


図２：各金属量のモデルについて得られた連星ブラックホールの有効スピンの累積規分布

これらの結果は，ブラックホールやその前駆体が伴星からの潮汐力以外にスピン角運動量を得ていなくても、合体するBBHの約16%は有効スピンが0.1以上になることを意味する。Wolf-Rayet星がcommon envelopeの直後には、無次元のスピン分布が平坦で等方的であると仮定すると、合体するBBHの有効スピン分布は、LIGOやVirgoによる重力波観測から推測されるものと似ていることを発見した。

参 考 文 献

[1] Abbott R., et al., 2021 Physical Review X, 11, 2

[2] Wang L., Spurzem R., Aarseth S., Nitadori K., Berczik P., KouwenhovenM. B. N., Naab T., 2015, Monthly Notices of the Royal Astronomical Society, 450, 4070

[3] Kroupa P., 2001, Monthly Notices of the Royal Astronomical Society, 322, 231

[4] Portegies Zwart S. F., McMillan S. L. W., 2000, Astrophysical Journal, 528, L17

[5] Fujii M. S., Tanikawa A., Makino J., 2017, Publications of the Astronomical Society of Japan, 69, 94

[6] Ziosi B. M., Mapelli M., Branchesi M., Tormen G., 2014, Monthly Notices of the Royal Astronomical Society, 441, 3703

[7] Piran Z., Piran T., 2020, Astrophysical Journal, Volume 892, 64

「東大情報基盤センターのスーパーコンピュータ」利用制度説明会

「計算＋データ＋学習」融合へ向けて

中島研吾

東京大学情報基盤センター

2021 年 10 月 8 日（金）にオンラインで開催された『「東大情報基盤センターのスーパーコンピュータ」利用制度説明会「計算＋データ＋学習」融合へ向けて』[1]の概要を報告する。

東京大学情報基盤センター（本センター）のスーパーコンピュータは様々な制度の元で利用することができる。本センターでは，利用者の皆様からは，電気代に相当する負担金を頂いているが，様々な公募制度を利用することにより，お得な使い方も可能である。本説明会は，下記に示すような制度，特に 10 月から課題募集が始まる 2022 年度 4 月開始の公募課題（HPCI，JHPCN，若手・女性，AI for HPC，企業利用）について詳しく紹介するために実施した。

- 一般利用，トライアル利用
- お試し利用，講習会
- HPCI，JHPCN
- 若手・女性，AI for HPC
- HPC チャレンジ
- 教育利用
- 企業利用

また，来年度も運用を継続する，Oakbridge-CX（OBCX，大規模超並列スーパーコンピュータシステム，2019 年 7 月運用開始），Wisteria/BDEC-01（「計算・データ・学習」融合スーパーコンピュータシステム，2021 年 5 月運用開始）の説明も利用事例を交えて実施した。スーパーコンピューティングは従来の計算科学・計算工学シミュレーションからデータ科学，機械学習・AI などのより広範囲の分野をカバーするようになり，「計算＋データ＋学習」融合による新しい問題解決手法が注目されている。Wisteria/BDEC-01，Oakbridge-CX は大規模シミュレーションに重点を置きつつも，「計算＋データ＋学習」融合を目指したシステムである。本利用制度説明会では，従来の利用分野とともに「計算＋データ＋学習」融合を考えておられる研究者，技術者の皆様に，本センターとしていかにお役に立てるか，情報をご提供できる機会になればと考え，実施した。本センターのスパコンの利用者でない方にも参加いただいた。合計 33 名の参加申込み（うち 22 名出席）があった。また，終了後は希望者に個別相談も実施した。

説明会の録画は，発表資料（PDF）とともに本説明会のホームページ（下記）でご覧いただける。当日参加出来なかった方は是非ご覧ください！

[1] https://www.cc.u-tokyo.ac.jp/events/seminar/20211008.php

JHPCN から「計算・データ・学習」融合を目指す

学際大規模情報基盤共同利用・共同研究拠点（JHPCN）公募説明会＠東大（オンライン）

中島研吾

東京大学情報基盤センター

2021 年 11 月 25 日（金）にオンラインで開催された『JHPCN から「計算・データ・学習」融合を目指す : 学際大規模情報基盤共同利用・共同研究拠点（JHPCN）公募説明会＠東大（オンライン）』 [1]の概要を報告する。

「学際大規模情報基盤共同利用・共同研究拠点（JHPCN）」[2]は、北海道大学，東北大学，東京大学，東京工業大学、名古屋大学、京都大学、大阪大学、九州大学にそれぞれ附置するスーパーコンピュータを持つ８つの施設を構成拠点とし，東京大学情報基盤センターがその中核拠点として機能する「ネットワーク型」共同利用・共同研究拠点であり，文部科学大臣の認定を受け，2010 年 4 月より本格的に活動を開始した。

JHPCN の主要な活動は，各拠点と共同で実施する，スーパーコンピュータ（スパコン）を使用した公募型学際的共同研究で，毎年 11 月から公募を開始し，1 月に提案書提出を締切り，その後の審査を経て，翌年 4 月より共同研究を実施する。提案書が採択された課題は，様々な計算機資源を原則無料で使用することが可能となる。大学・研究機関の教職員の他，企業の研究者，技術者も応募することができる。

スパコンの用途は従来の計算科学シミュレーションから，データ解析・処理，機械学習・AI と広がりを見せている。またこれら「計算・データ・学習」の融合に基づく新しい計算科学の開拓は，エクサスケール時代のスパコンの能力を最大限活用し，科学的発見を持続的に促進するとともに，サイバー空間（仮想空間）とフィジカル空間（現実空間）を高度に融合したシステムを形成し，Society 5.0 の実現に大きく貢献すると期待されている。

東京大学情報基盤センターでは，2015 年頃からこのような状況を考慮して，「計算・データ・学習」の融合を目指した研究開発，システム整備を継続して実施してきた。2021 年 5 月には，「（計算・データ・学習）融合スーパーコンピュータシステム（Wisteria/BDEC-01） 」[3]が運用を開始し，また，2019 年 6 月より「（計算＋データ＋学習）融合によるエクサスケール時代の革新的シミュレーション手法」に関する研究開発を科研費基盤研究（S）（代表 : 中島研吾（東京大学情報基盤センター））として開始し，革新的ソフトウェア基盤「h3-Open-BDEC 」[4]を整備しており，現在は Wisteria/BDEC-01 上での検証，研究開発を継続して実施している。h3-Open-BDEC の開発は，JHPCN を構成する北海道大学，名古屋大学，東京工業大学，九州大学，及び産学官の様々な分野の研究者，技術者との協力のもと実施している。

東京大学情報基盤センターは，2022 年度の JHPCN 共同研究課題では，Wisteria/BDEC-01 とともに，「大規模超並列スーパーコンピュータシステム（Oakbridge-CX ，OBCX） 」[5]を提供し，

1 https://www.cc.u-tokyo.ac.jp/events/seminar/20211125.php
2 https://jhpcn-kyoten.itc.u-tokyo.ac.jp/ja/
3 https://www.cc.u-tokyo.ac.jp/supercomputer/wisteria/service/
4 https://h3-open-bdec.cc.u-tokyo.ac.jp/
5 https://www.cc.u-tokyo.ac.jp/supercomputer/obcx/service/

広範囲な課題に対応するべく，皆様をサポートしていく予定である。これら東京大学情報基盤センターのシステムを使用して，JHPCN の公募型学際的研究課題への応募を検討している皆様を対象に，各システムの概要，現在実施中の共同研究課題の事例紹介，提案に向けた共同研究体制構築に向けた説明会を企画，実施した。合計 19 名の参加申込み（うち 14 名出席）があった。また，終了後は希望者に個別相談も実施した。

説明会の発表資料（PDF）は，本説明会のホームページ（下記）でご覧いただける。当日参加出来なかった方は是非ご覧ください！

https://www.cc.u-tokyo.ac.jp/events/seminar/20211125.php

第 165 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

2021 年 10 月 12 日（火）、第 165 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：3 名、企業：1 名、その他：1 名で、合計 7 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、Oakforest-PACS 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。

- 10 月 12 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/UC2CHaGp1AO-vqRlV7wmU0-w/videos）にアップロードした（演習：https://www.youtube.com/watch?v=F97vZZvI1BQ, 座学：https://www.youtube.com/watch?v=Seo5Ht8XAWI）のでいつでも視聴できるようになっています。

7 名中 6 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 1 年の方が 2 名と最多でしたが、最長で 6 年と回答された方もおり幅広く分布していました（図 1）。一方で並列プログラミング経験については、並列プログラミング入門ということもあり、まったくない方が 4 名と大半を占めていました。またあると答えられた方も 1 年未満であるとのことでした. 普段使用しているプログラミング言語については、Python や C/C++及び Fortran でほぼ同数でした（図 2）。

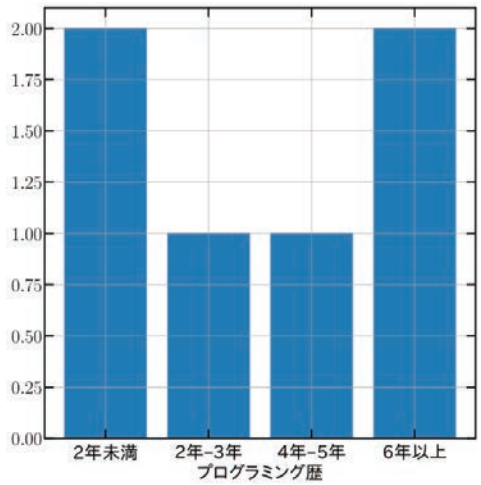


図 1：参加者のプログラミング経験

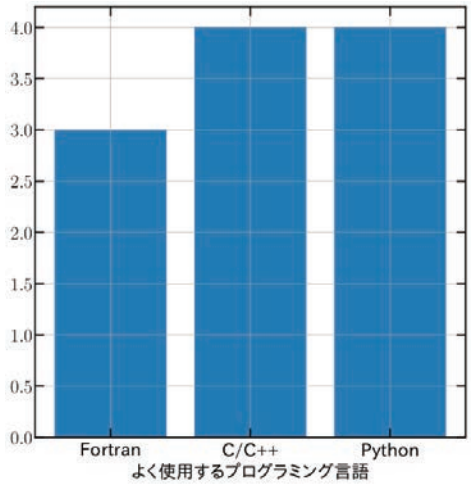


図 2：普段使用しているプログラミング言語

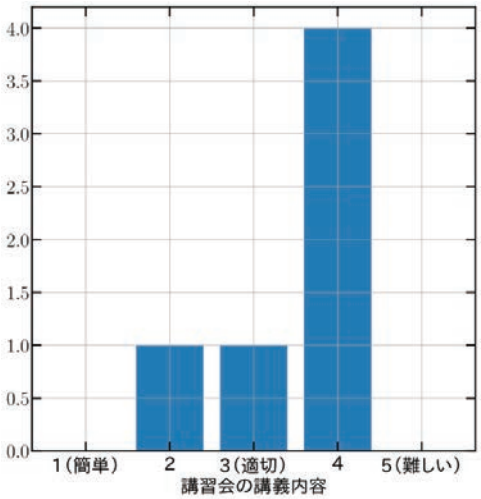


図 3：講習会の講義内容（プレゼン）

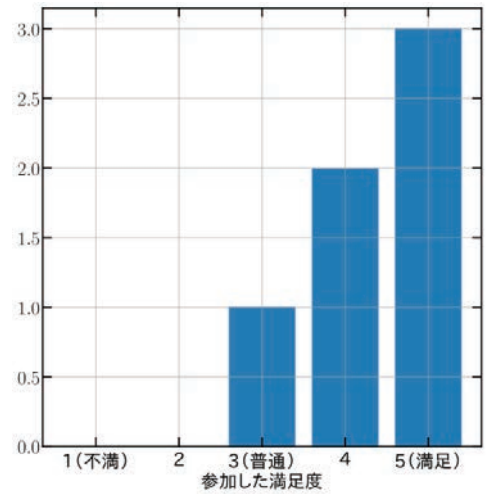


図 4：講習会に参加した満足度

講習会の難易度設定については、難しめである 4 と回答された参加者が最多であり、現地

開催での講習会よりも難しく感じる方が増えているようです（図 3)。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(平均値は4.3)。

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

- 遠方から参加可能なこと
- 講習会会場への移動の必要がないこと
- 質問に対し即座に回答してもらえたこと（2名）
- Slackで質疑応答の履歴を残してもらえるため，後で復習に使いやすいこと
- スライドが見やすい
- 自分の環境で演習ができる
- 与えられた課題の難易度は適度に考えて納得できたレベルでちょうど良かったが，オンラインは独りなのでそれを自分のペースで行えたこと

で、悪かったことについての回答は

- 質疑のやり取りのレスポンスが悪い（これはZoomとSlackを併用した今回の方式に不満があるという意味ではなく，対面に比べるとどうしても意思疎通が難しくなるという意味）

というものでした。会場まで移動する必要がなくなったため遠隔地からの参加が容易になったというメリットを回答される方が多かったです。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。

- 以前，他大学にて行われたMPI講習会に参加したことがあったが，その時よりも非常にわかりやすい内容であった.
- 講義資料・ご説明ともに非常にわかりやすく、今後のMPIの学習を進める上でとてもためになる講習会だったと感じております。MPIの事前学習が不十分で、午後の演習は苦戦してばかりだったため、本日の資料と演習用のコードを用いて、これから復習しようと思います。本日は丁寧なご指導、本当にありがとうございました。
- 最後の拡散方程式の例が、問題内容をあまり理解できずに課題が進んだように感じ、あまりついていけなかった。実例なので、できれば丁寧にやってほしかった。
- 拡散方程式の課題になった途端、系や実装に関連しているがMPIとは離れた事項を頭に入れる必要があり、演習にスムーズには入れなかった。後でプログラムの穴埋め以外の部分も見てみたい。

同様の講習会があれば、「また受けたい」という回答が3名、「どちらともいえない」が3名で、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 166 回お試しアカウント付き並列プログラミング講習会

「MPI 上級編」実施報告

塙　敏博
東京大学情報基盤センター

2021 年 10 月 18 日（月）、第 166 回お試しアカウント付き並列プログラミング講習会「MPI 上級編」が開催されました。MPI 上級編は、毎年 1 回ずつの開催で今回が 5 回目となりました。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る本講習会ですが、今回は新型コロナウイルス感染症対策のために Zoom 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。

受講者は、大学・研究機関教職員： 1 名、大学院学生：2 名、企業の方：1 名、その他：1 名、参加者合計：5 名、でした。

今回からは、Wisteria/BDEC-01 スーパーコンピュータシステムの 1 ヶ月有効なお試しアカウントが与えられ、MPI(Message Passing Interface)の高度な機能を用いたプログラミングに関する講習会を 1 日間で実施しました。

当日のプログラムを、以下に掲載します。

- 10 月 18 日（月）
  - 10：00 - 11：20　MPI 概要、Wisteria/BDEC-01 で使える MPI 実装
  - 11：30 - 12：30　ノンブロッキング通信（演習）
  - 13：30 - 14：30　派生データ型、MPI-IO（演習）
  - 14：40 - 16：10　コミュニケータ、マルチスレッド（演習）
  - 16：20 - 17：30　片側通信（演習）

4 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。

MPI 上級だけあって、プログラミング経験については、3 年以上、40 年を超える方もいらっしゃいました。使用しているプログラミング言語については、Python がトップで、次が C と C++（複数回答可）となっています。

主要な項目の集計結果を以下に示します。

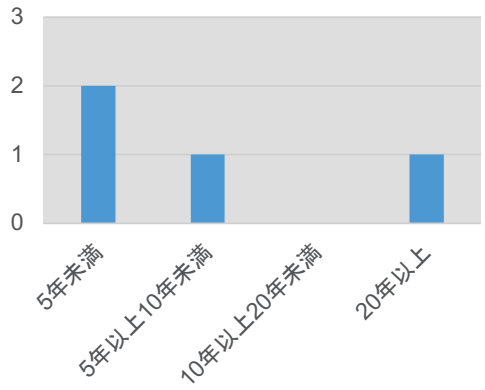


図 1 プログラミング経験

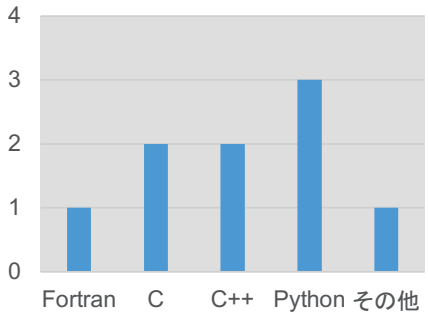


図 2 普段使用するプログラミング言語

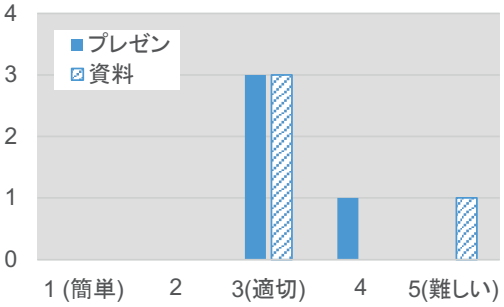


図 3 講習会の内容

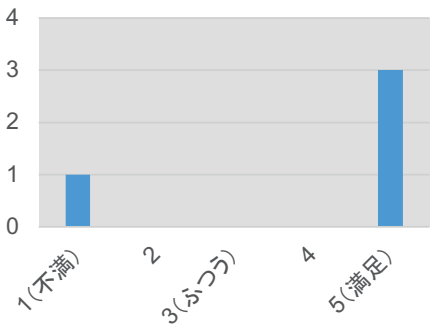


図 4 講習会参加の満足度

図 4 より、顧客満足度の平均値は 4 でした。

また、以下の感想をいただきました。

- 資料以外で、コマンドライン打ってやった内容が多いため、収録した映像を公開していただければ幸いです。
- 各講習会の動画が HP で公開されるようになり、リアルタイムで参加できなくても事後に学べてありがたいです。その際に、サンプルプログラムも何らかの方法で入手できるよう、対応をお願いしたいです。
- 扱うテーマは MPI に関して多岐にわたっていたが，いずれも単純な例での演習が提供されており，その機能を中心に理解するのに役立った．特に，図示も含め片側通信の

違いについてが分かりやすかった.

- MPI 基礎を受講している人も含まれるだろうから，システムについての説明はさほど長くても良いように感じた.

Wisteria/BDEC-01 を対象にしましたが、Odyssey と Aquarius でアーキテクチャが大きく異なりますので、双方をカバーするとかなりのボリュームになります。1 日で実施している関係で、演習時間については不足気味で、一通りの説明に終わったものもありました。今後も改善を続けていきたいと思います。

一方で、ホームページで資料を公開しておりますし、YouTube で講習会録画も公開していますので、関心のある方はごらんください。

同様の講習会があれば、「また受けたい」という回答が 3 名、「どちらともいえない」が 1 名で、その他の講習会にも期待されていることが伺えます。

平成 24 年 4 月から、当センターのスーパーコンピュータシステムを利用する企業利用者向けトライアルユース制度（パーソナルコース相当）では、お試しアカウント付き講習会の受講が義務づけられています。企業の方でトライアルユース制度（パーソナルコース相当）をご利用の方は、本講習会の日程について事前にご確認ください。

詳細および講習会への申込みは、以下のホームページでご確認ください。

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

以上

第 167 回お試しアカウント付き並列プログラミング講習

「OpenACC と MPI によるマルチ 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，2021 年 10 月 19 日にオンライン会議システム Zoom を用いて「OpenACC と MPI によるマルチ GPU プログラミング入門」が開催されました。 本講習会は，東京大学内および学外における本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や技術者の方が参加することができます。本講習会では，GPU に焦点をあて，GPU 搭載スパコンで大規模な数値計算を実現するために必須となるマルチ GPU プログラミングについて学びます。GPU は，NVIDIA 社の Tesla GPU（Pascal アーキテクチャ）を対象とし，プログラミング環境としては既存コードに指示文（ディレクティブ）を追加することで GPU 化できる OpenACC を用います。GPU 間の並列化は MPI を利用します。実習では，基礎的なプログラムを通して，OpenACC による GPU コードの作成，OpenACC と MPI による複数 GPU を用いたプログラムの作成方法を学びます。実習には，2021 年 5 月 14 日より運用を開始した Wisteria/BDEC-01 の Aquarius ノードを使用します。本講習会のスケジュール は表 1 の通りです。前回までは午前・午後の 1 日の講習会でしたが、「GPU プログラミング入門」と重複する部分も多かったことから、今回からスケジュールを見直し、午後半日の講習会としております。講習会の内容の詳細や講習会で使用した資料は，講習会の Web ページ[1] に掲載しておりますので，そちらをご覧ください。受講者には実習で使用した Wisteria/BDEC-01 を受講後 1 ヵ月間利用できるお試しアカウントが与えられます。 今回の講習会では，合計 8 名の事前申込者があり，そのうち 7 名が受講しました。受講者の内訳は，大学院学生：3 名，大学研究機関職員： 1 名，企業の方 : 2 名，その他 : 1 名でした。講習会終了後にアンケートを実施した質問項目と回答の人数分布 は表 2 の通りです。

表 1 スケジュール

時間	内容
13:00〜13:30	スパコンの使い方など
13:30〜14:30	OpenACC+MPI 混合プログラミング（座学）
14:45〜17:00	OpenACC+MPI 演習

[1] http://nkl.cc.u-tokyo.ac.jp/support/kosyu/167/

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

第 168 回お試しアカウント付き並列プログラミング講習会

「科学技術計算の効率化入門」実施報告

河合 直聡

スーパーコンピューティングチーム

2021 年 10 月 26 日（火）、第 168 回お試しアカウント付き並列プログラミング講習会「科学技術計算の効率化入門」が、オンラインにて開催されました。

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。

受講者は、大学院学生、研究機関研究員、企業の方がそれぞれ 1 名ずつの計 3 名でした。

1 カ月有効なお試しアカウントが与えられ、Oakbridge-CX スーパーコンピュータシステムの利用方法、科学技術計算ライブラリ利用に関する演習、シミュレーションの効率化に関する講習が、終日の日程で行われました。

当日のプログラムを、以下に記します。

10 月 7 日（水）

13:00–13:15	システム紹介
13:15–14:15	スパコンと線形計算ライブラリ（BLAS, LAPACK）
14:15–14:30	休憩＆質問
14:30–15:45	Xcrypt を用いたジョブ並列処理
15:45–16:30	実習＆質問

講習会終了後にアンケートを実施しました。参加者の 3 名全員から、講習会に関するアンケートをご提出いただきました。表 1 は質問項目と回答（5 段階評価）の人数分布です。今回はプログラミング経験が無い方はおらず、全員 Fortran 経験者ですが、並列プログラミングの経験は 1〜2 年と短い方々に参加いただけました。講習会の時間は参加者全員から適切との回答を得ており、また、講義内容、配布資料の内容の評価は適切かまたはやや簡単との評価でした。サンプルプログラムの内容に関してもやや簡単からやや難しいまで 1 票ずつの回答であり、全体的な満足度は全員満足(5)の評価をいただきました。

以下のご意見を頂きました。

■Zoom によるオンライン講習会で良かったこと

・音声が明瞭

・時間が有効活用できる
・地方在住のため、時間と費用の節約になったこと

■Zoom によるオンライン講習会で悪かったこと
・スパコン本体も見てみたい。

■講習会全般に対する意見・要望など
・Xcrypt のような並列処理をスパコンに限らず手持ちサーバーでも利用したかったので、大変有用でした。ライブラリ利用法も使ってみます。

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			3		
(b) 講習会講義内容（プレゼン）	簡単⇔難			3		
(c) 配布資料内容	簡単⇔難		1	2		
(d) サンプルプログラム内容	簡単⇔難		1	1	1	
(e) 満足度（平均 4.0）	不満⇔満足					3

以上

第 169 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

2021 年 11 月 1 日（月）にオンライン開催した第 169 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））（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））」[1]について紹介する。

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（ディレクティヴ）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，「有限体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[2]。第 154 回講習会（2021 年 5 月 31 日（月）実施）[3]では，2021 円 5 月 14 日に運用を開始した Wisteria/BDEC-01（Odyssey）[4]を使った最初の講習会として企画し，第 151 回の内容のうち，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て実施したところ，大変好評であったため，今回も同様のカリキュラムにて実施した。

アンケート結果（申込者 5 名，受講者 4 名，回収 3 名）を表 2 に示す（満足度平均 4.67）

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

[1] https://www.cc.u-tokyo.ac.jp/events/lectures/169/
[2] http://nkl.cc.u-tokyo.ac.jp/21s/
[3] https://www.cc.u-tokyo.ac.jp/events/lectures/154/
[4] https://www.cc.u-tokyo.ac.jp/supercomputer/wisteria/service/

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			3		
(b) 講習会講義内容（プレゼン）	簡単⇔難		2	1		
(c) 配布資料内容	簡単⇔難		1	2		
(d) サンプルプログラム内容	簡単⇔難		1	2		
(e) 満足度（平均 4.67）	不満⇔満足				1	2

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/169）。

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ー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 原稿枚数には特に制限はありませんが，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，A 4 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://ofp-www.jcahpc.jp/ (Oakforest-PACS)

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/#SOUDAN
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 (Wisteria-O/A)	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
Oakforest-PACS スーパーコンピュータシステム	ofp.jcahpc.jp 以下のホストの何れかに接続します※ ofp0{1-6}.jcahpc.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）

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ニュース
Vol.24 No.1(2022.1)

目　　次