

## [2022]九州大学情報統括本部年報 : 2022年度

<https://hdl.handle.net/2324/7157415>

---

出版情報 : 九州大学情報統括本部年報. 2022, pp.1-, 2023-11-01. Information Infrastructure Initiative, Kyushu University

バージョン :

権利関係 :



## 第8章 イベント紹介

### 8.1 「 $\alpha$ xSC2018Q ゲームとスーパーコンピュータに関するシンポジウム」

これは、「なにか×スーパーコンピュータ (SC)」というテーマを掲げ、その「なにか」の分野で、今までスーパーコンピュータとの関わりがあまりなかった方、および、既にスーパーコンピュータを活用されている方、それぞれにご自身の研究や業務の内容を講演して頂き、さらに、それらを踏まえた講演者および参加者の意見交換を行うシンポジウムです。

今回は、その「なにか」を「ゲーム」とし、4名の講演者の皆様に、将棋や囲碁のような対戦ゲーム、さらにモバイルオンラインゲームや社会問題の解決を目的とするシリアスゲームを題材に、現在の取り組みや将来の展望、さらにスーパーコンピュータ活用の可能性について、ご講演いただきました。また、今後のゲームとスーパーコンピュータの関わりについての活発な議論を行うことが出来ました。

開催日時：2022年4月8日(火) 13時20分～17時30分

場 所：オンライン開催

主 催：九州大学情報基盤研究開発センター

共 催：京都大学学術情報メディアセンター

#### 【プログラム】

13:20～13:30 開催挨拶

13:30～14:15

「大規模な計算資源の強いゲームプレイへの貢献」

美添 一樹 (九州大学情報基盤研究開発センター 教授)

深層学習と探索を用いた AlphaGo が囲碁棋士を倒したことは記憶に新しい。本講演では、深層学習モデルのトレーニングと並列探索の2つの観点から大規模な計算資源がどのように強いゲームプレイに貢献したのかを解説する。さらに強いゲーム AI の研究から得られた手法が他の分野にもたらしつつある貢献について、講演者の研究にも触れながら紹介する。

14:15～15:00

「機械学習とスパコンが切り拓くモバイルオンラインゲームの未来」

濱田 直希 (KLab 株式会社)

昨今のモバイルオンラインゲームの世界市場規模は7兆円に達し、なお急速に拡大を続けている。それに伴ってゲームの開発・運用も急速に大規模化しており、効率化が不可欠になっている。KLab 株式会社では、ゲームの開発・運用で生じる豊富なデータを活用して、様々な機械学習に取り組んでいる。本講演では、九州大学櫻井研究室との共同研究による、リズムアクションゲームの譜面生成について紹介する。スパコ

ンで深層生成モデルを訓練し、その結果をクラウドにデプロイすることで、研究成果を瞬時に利用可能にした。最後に、機械学習とスパコンによって可能となる新たなゲームについて展望を述べる。

15:15 ~ 16:00

「シリアスゲームプロジェクト in メタバースは可能か!？」

松隈 浩之 (九州大学大学院芸術工学研究院 准教授)

シリアスゲームとは社会問題の解決を目的に据えたデジタルゲームであり、2009年から高齢者のリハビリヘルスケア支援をテーマにシリアスゲームプロジェクト (SGP) を推進しております。また、2020年以降、コロナ禍で直接会うことに制約がかかる社会情勢にあわせて、大橋キャンパスを3DCGで作成し、その中で謎解き脱出ゲーム『オルタグ-大橋キャンパスを目指せ』をプレイするという(流行の)メタバース的な取り組みを実験的にこなっております。本講演では「ゲーム」という切り口で「シリアスゲームプロジェクト」および『オルタグ-大橋キャンパスを目指せ』の解説をするとともに、メタバース空間でシリアスゲームの実施は可能か、その場合にどれくらいのコンピュータによる計算が必要になるのだろうかという内容でお話しできればと思います。

16:00 ~ 16:45

「ゲームの解決」

田中 哲朗 (東京大学情報基盤センター 准教授)

特定のゲームを対象にしたゲームプログラミングの研究では、「人間のトッププレイヤーを倒す」「ゲーム解決」という2つの大きなマイルストーンがある。この2つのマイルストーンを達成するためには、大量の計算機資源が必要となる。本講演では、まず、講演者が開発に参加したGPS将棋の話を中心に、過去にコンピュータプレイヤーがどのように人間のトッププレイヤーを超えてきたかを紹介する。また、次のマイルストーンである「ゲーム解決」について。過去に解決されたゲームと、今後の展望について解説する。

16:45 ~ 17:30 総合討論

## 8.2 「先駆的科学計算に関するフォーラム 2022」

2022 年 4 月 25 日 (月) にオンライン (Microsoft Teams) を用いて、最新のハイパフォーマンスコンピューティング (HPC) に関する先端的研究成果の紹介や研究用計算機システム利用者の情報交換の場として、標記フォーラムを開催しました。フォーラムには技術スタッフ、センター運用関係者も出席し、熱心な質疑応答が交わされました。

### 【フォーラムプログラム】

13:00 開会の挨拶

13:05 ~ 13:25

北村 拓也 (長崎大学)

「流体乱流の大規模数値計算と数理モデル化」

13:25 ~ 13:45

町田 真美 (国立天文台)

「富岳を用いた AGN ジェット伝搬の 2 温度磁気流体シミュレーション」

13:45 ~ 14:05

木野 康志 (東北大学)

「エキゾチック原子分子の反応と構造の理論計算」

14:05 ~ 14:25

宇都 卓也 (宮崎大学)

「糖鎖高分子やイオン液体における構造機能相関の解明を目指した  
分子シミュレーション研究」

(休憩)

14:40 ~ 15:00

伊藤 一秀 (九州大学)

「健康影響を可視化する in silico 人体モデルの開発と環境設計への展開」

15:00 ~ 15:20

濱田 直希 (KLab 株式会社)

「機械学習を用いた譜面制作支援の研究」

15:20 ~ 15:40

村田 健史 (情報通信研究機構)

「広域 L2VPN 通信網を用いた分散型データ公開システムの検証」

(休憩)

15:55 ~ 16:20

[先端的計算科学研究プロジェクト成果報告 I]

深沢 圭一郎 (京都大学)

「Ito システムにおけるコード結合フレーム CoToCoA を用いた

宇宙プラズマ連成計算シミュレーションの計算・電力性能評価」

16:20 ~ 16:45

[先端的計算科学研究プロジェクト成果報告 II]

高橋 裕介 (北海道大学)

「はやぶさ型再突入カプセルの動的不安定メカニズムの解明と低減化」

16:45 ~ 17:00

自由討論・閉会の挨拶

### 8.3 「国際会議 SC22 Exhibition における研究開発展示」

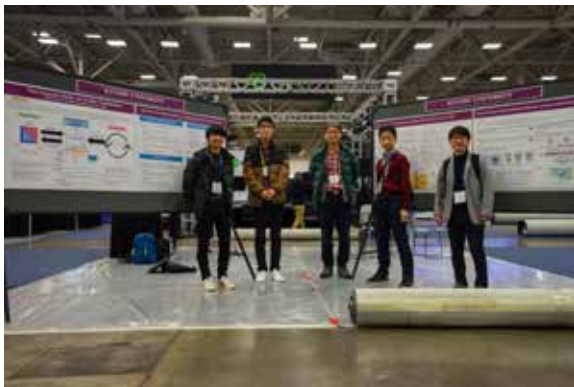
#### ■ SC22

SC は毎年米国で開催されている国際会議で、参加者は 12,000 ～ 13,000 人と非常に多く、また参加する研究機関および企業も世界中から集まる、情報関連分野では世界最大規模のイベントです。SC22 は、米国 Texas 州の Dallas 市で開催されました。表 1 に SC22 の会議場や会期、および九州大学ブースの情報を記載します。

表 1 : SC22 概要および九州大学ブース情報

SC22 Web Site	<a href="http://sc22.supercomputing.org/">http://sc22.supercomputing.org/</a>
場所	Kay Bailey Hutchison Convention Center Dallas (Dallas, TX, USA)
SC22 期間	2022 年 11 月 13 日 (日) ～18 日 (金)
Exhibition 期間	2022 年 11 月 14 日 (月) ～17 日 (木)
九大ブース	No. 3941 (20x20 フィート)

SC の会議トピックは大規模計算や高性能の計算機、ネットワークやストレージと多岐にわたっています。SC には主な行事として、研究論文の Technical Program、最新技術に関する Tutorial、テーマを絞って開催される Workshop や BoF (Birds of Feather)、さらに、企業や研究機関が展示ブースを並べて商品や研究成果を紹介する Exhibition があります。当センターは、改組前の情報基盤センターであった 2003 年から、ほぼ毎年、九州大学の展示ブースを設営し、九州大学における計算科学および計算機科学に関する研究成果や、稼働している計算機資源などを紹介してきました。しかし COVID-19 の影響で SC20 はオンライン開催となり、SC21 はハイブリッド開催だったものの海外出張が厳しく制約されたため、2 年間、現地での展示が行えませんでした。今回は感染状況の好転にとともに、久しぶりの現地での展示となりました。



展示場の様子 (九大展示ブース)



SC22 オープニング前のメインホールの様子

■ 展示ポスター

九州大学ブースでは、下記の 7 種の展示を行いました。

(1) Research Institute for Information Technology, KYUSHU UNIVERSITY

(2) Supercomputer System “ITO” - 10 PFLOPS Hybrid Supercomputer

Total	Performance	Memory	Memory BW	Power	# Racks
	<b>10.43 PFLOPS</b>	<b>441 TB</b>	<b>960.2 TB/s</b>	<b>2667.6 KVA</b>	<b>74 racks</b>
	CPU: 7.72 PFLOPS GPU: 2.71 PFLOPS	DDR4: 433TB HBM2: 8TB	DDR4: 585.5TB/s HBM2: 374.7TB/s	estimated maximum with HPL, includes cooling	

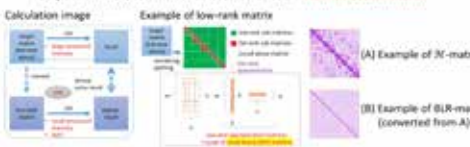
(3) Advanced GPU computing: utilizing MIG and RT core for HPC

### Advanced GPU computing: utilizing MIG and RT core for HPC

[This work was partially supported by JSPS KAKENHI Grant Numbers 21H03447 and 21K19763.]

#### Low-Rank matrices on GPU

- Although Moore's law is approaching the limit, the growth of computational performance supports the improvement of various sciences. The growth of the speed and capacity of memory is not catch up with calculation performance. **Low-Rank approximation methods** have attracted attention as a solution to this issue.
- Low-Rank approximation methods use approximated matrices instead of raw matrices. Various applications can be solved fast with a small amount of memory within the acceptable error.
- How about it on GPU?
  - high performance of GPU is yielded by a lot of cores
  - high parallelism (very long loop) is required, but Low-Rank matrices require numerous small matrix calculations. **How to accelerate it?**



#### $\mathcal{H}$ -matrix on GPU

- To accelerate Hierarchical matrix ( $\mathcal{H}$ -matrix) - vector multiplication in Bi-CGSTAB method on GPU, batched calculation is employed. We succeeded to obtain high performance than CPU.
- Satoshi Ohshima, et al.: Optimization of Hierarchical matrix computation on GPU, SCFA 2018, LNCS 10776, DOI 10.1007/978-3-319-69953-0\_16, 2018
- Satoshi Ohshima, et al.: Optimization of Numerous Small Dense-Matrix-Vector Multiplications in H-matrix Arithmetic on GPU, MCSoc 2019, DOI 10.1109/MCSoc.2019.00009

#### Block Low-Rank QR factorization (BLR-QR) on Multi-Instance GPU (MIG)


- BLR is a simple case of  $\mathcal{H}$ -matrix, suitable for distributed computing.
- Batched computation is a hopeful solution, but it requires major code rewriting. To obtain high performance without major code rewriting, we employed MIG feature. With MIG, we can use GPU libraries in limited GPU and it is useful in calculating numerous small calculations on GPU.
- The latest result will be shown at POCAT 2022 (this December at Sendai).
- Satoshi Ohshima, et al.: QR Factorization of Block Low-Rank Matrices on Multi-Instance GPU, POCAT2022, <https://www.hpc.is.tohoku.ac.jp/pocat2022/>

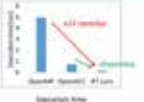
---

#### Utilizing RT core for scientific applications

contributed by Shinya Hashimoto (Nagoya Univ.)


- Ray-tracing is an important algorithm in various scientific applications. Because current GPUs have ray-tracing acceleration units, we want to utilize them for scientific applications.
- Our current target application is radio wave propagation loss calculation. We succeeded to accelerate its simple problem settings by utilizing RT core of NVIDIA GPUs.
- We have already published its result in Japanese domestic research reports. Please wait for the English report at the future conference.





- OptiX obtained the fastest execution time.
- OpenMP and OpenACC use SBT algorithm. RT core uses BVH algorithm. (By using OptiX, BVH is applied automatically. The ease is also merit of OptiX.)

Satoshi Ohshima  
Associate Professor  
ohshima@cc.kyushu-u.ac.jp  
<http://weh.net>



[This work was partially supported by JSPS KAKENHI Grant Numbers JP21K19763.]

(4) CoToCoA: Code To Code Adapter for Coupled Simulations

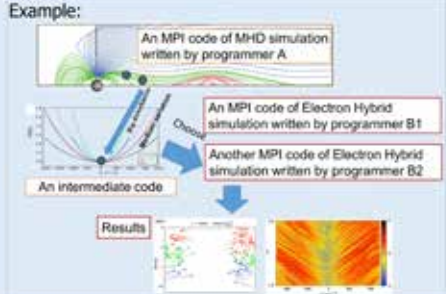
### CoToCoA: Code To Code Adapter for Coupled Simulations

Takanori NANRI (Kyushu Univ.), Yuta KATORI (Tohoku Univ.), Keiichi FUKAZAWA (Kyoto Univ.) and Yohji MIYAKE (Kobe Univ.)

#### Motivation:

Connect MPI codes for coupled simulations with minimal modification to original codes

Example:



#### CoToCoA (Code To Code Adapter)

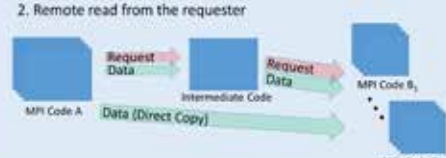
<https://github.com/tnanri/cotocoa>

Goal 1: Minimize modification to existing MPI programs  
Connection via a coupler program that controls connection

Goal 2: Asynchronous execution  
Requester-Worker model.

Goal 3: Low overhead of data transfer

Two styles of data-transfer:  
1. Attach an array to a request  
2. Remote read from the requester



---

#### Usage of CoToCoA

**Requester:**

- Compute
- EXPOSE data area, or ATTACH data to a request
- SUBMIT a request to Coupler

```

MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
MPI_Comm_size(MPI_COMM_WORLD, &size);
// ...
MPI_Request request;
MPI_Request(&request);
MPI_Sendrecv(...);
MPI_Wait(&request, MPI_STATUS_IGNORE);
        
```

**Coupler:**

- POLL a request from Requester
- Collect data, calculate, and choose a worker
- SUBMIT request to the worker

```

MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
MPI_Comm_size(MPI_COMM_WORLD, &size);
// ...
MPI_Request request;
MPI_Waitany(&request, &index, MPI_STATUS_IGNORE);
MPI_Recv(...);
// ...
MPI_Send(...);
        
```

**Worker:**


- POLL a request from Coupler
- Collect data
- Compute

```

MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
MPI_Comm_size(MPI_COMM_WORLD, &size);
// ...
MPI_Request request;
MPI_Waitany(&request, &index, MPI_STATUS_IGNORE);
MPI_Recv(...);
// ...
MPI_Send(...);
        
```

---

#### Evaluation:



Elapsed time of a request from "sendreq" to "complete" with three data-transfer styles:

- CoToCoA attach: Attach to a request
- CoToCoA read: Remote read
- MPI: Use MPI\_send and \_recv

Environment: ITD A (Kyushu Univ.)  
- Intel Xeon, Mellanox InfiniBand FDR, Redhat Enterprise 7, Intel MPI, 4 nodes, 4 proc / node  
- 8 proc for requester, 1 proc for coupler and 6 proc for worker

(5) Time Sequence Buffer for In-Situ Visualization

### Time Sequence Buffer for In-Situ Visualization

MAENO Yuta, NANRI Takeshi (Kyushu University)

#### Motivation

In-situ visualization requires storing the results of simulations in a buffer in chronological order. Generally, the size of the buffer is not sufficient to save all results. In this work, we develop the Time Sequence Buffer that runs as one of the MPI processes on multiple nodes running the simulation. For efficient use of the Buffer, the Visualizer should be able to erase arbitrary data and save it to disk. For this purpose, we study slot management technology for the Buffer.

#### Simulation

Loop

- Visualize the number of calculation results using **Unity**. The Unity facilitates Visualization and on-screen manipulation.
- Each button on the screen has its own function to manipulate data. when a button is pressed, each request is sent to the buffer side.
  - Prev: Return to the previous time step
  - Next: Proceed to the next time step

#### Definition

- Stores data with time step along the way in the simulation

#### Function

- Continues to receive and store data for as long as there is room in the buffer
- Transmits, erases, or saves data according to the requests from visualizer

#### Implementation

- Uses hash-tables to search for empty slots and requested steps

- While the simulation loops until all calculations are completed, it sends calculated data to free buffer, and it stops if there is no room in the buffer.
- Processes huge amounts of calculations on multiple nodes.
- Target: simulation on dynamic systems
  - thermal flow of steel plate
  - weather forecast

(6) Dynamic optimization technique in tiering mechanism for hybrid storage

### Dynamic optimization technique in tiering mechanism for hybrid storage

OKAMURA Yutaro (Kyushu Univ.), OKAMURA Yutaro (National Institute of Informatics), NANRI Takeshi (Kyushu Univ.), okamura.yutaro.100@nii.ac.jp

#### Memory / Storage Hierarchy

Currently, hierarchical storage such as NVDIMM, SSD, HDD, and memory are used in computer systems. They each have the following characteristics.

- DRAM**
  - Volatile: Data is lost when the computer is turned off.
  - Very fast and accessible, but very expensive.
- NVDIMM**
  - Non Volatile: No data is lost when the computer is turned off.
  - Accessible at speeds equivalent to DRAM.
  - Byte-by-byte access, as opposed to lock-by-lock access for SSDs and HDDs.
  - Very expensive per capacity compared to SSDs and HDDs.
- SSD**
  - Faster than HDD with electrical read/write.
  - More expensive than HDD and limited number of writes.
- HDD**
  - Low price per capacity and large capacity.
  - Read/write is physical and slow.

#### Tiering mechanism for hybrid storage

- One system that enables hybrid storage through tiering is Automated Tiered Storage with fast Memory and slow Flash storage(ATSMF)[2].
- Partition storage into sub-LUN units and monitor IO accesses by these units.
- Migration of target area according to the predicted remained time of IO concentration.
- Predicted remained time analyzes the replay in advance and uses static values fixed by workload.

#### Hybrid Storage

Hybrid storage[1] is a method of combining devices with different performance(NVDIMM and SSD, SSD and HDD). They are classified into caching and tiering according to the algorithms used to exchange data between devices.

- Caching**
  - Use fast device as cache for slow device.
  - Performance is highly dependent on access patterns.
  - volume depends on slow device.
- Tiering**
  - Migration between devices of different performance to improve average throughput.
  - volume is available for the total of all devices.

#### Dynamic optimization technique in tiering mechanism for hybrid storage

- ATSMF performs migration based on the performance of NVDIMMs and SSDs, so if HDDs are targeted, operation is unstable.
- Need to analyze workload characteristics in advance to determine migration thresholds.
- Build ATSMF-Dynamics, which dynamically changes the migration decision criteria that are currently determined statically, to accommodate more generic workloads.

(7) Toward large-scale quantum computer system architecture

■ おわりに

SC22 は、COVID-19 の影響か全体的に参加者が少なく、展示会場の参加者数も SC19 に比べると大幅に減少した印象でした。その中で、九大ブースのポスター内容や九大センターの計算環境に興味を持った参加者と、議論や情報交換を行えました。また、他ブース訪問や Technical Program、Birds of Feather、Workshop 等への参加を通じて、九大ブース参加者も、それぞれスーパーコンピュータ、ネットワーク、ストレージ等の最新の研究動向や技術について知見が得られました。

次の SC23 は、米国 Colorado 州の Denver 市にある Colorado Convention Center にて開催予定です。SC23 でも九州大学ブースとして展示予定です。

表 2 : SC23 概要および展示予定ブース

SC23 Web Site	<a href="http://sc23.supercomputing.org/">http://sc23.supercomputing.org/</a>
場所	Colorado Convention Center (Denver, Colorado, USA)
SC23 期間	2023 年 11 月 12 日 (日) ~17 日 (金)
Exhibition 期間	2023 年 11 月 13 日 (月) ~16 日 (木)
九大ブース	No. 237 (20x20 フィート)