# SS 研科学技術計算分科会 2021 年度会合 分科会レポート

「富岳」NEXTへの挑戦　～現在から未来へ～

2023 年 1 月 20 日

大阪大学大学院　谷口昂平

## はじめに

2023 年 1 月 20 日，科学技術計算分科会 2022 年度会合が開催された．今年度のテーマは ”「富岳」NEXTへの挑戦　～現在から未来へ～” であった．現行の「富岳」は2014年から開発が始まり, 2021年3月の運用開始後は，Top 500 や他のランキングでも結果を残している．富岳の開発から運用開始まで7年要したことを鑑みると, 今日から7年後の2030年に運用されているであろう「富岳」の後継システムについて考え始める必要がある.

本会合では, 2030年を見据えた「富岳」の後継システムを議題として, 設計や運用など様々な観点から 5 つの講演が行われた．本会合は COVID-19 の影響によりオンライン開催だった前回とは異なり, 3 年ぶりの現地開催を含むハイブリッド開催となった. 現地参加者は約 45 名であり, オンライン参加者が約 40 名であった. 2030年の「富岳」の後継システムという難しい議題でありながら, 現地参加, オンライン参加問わず活発な議論が交わされた.

以降は各講演の概要と議論の内容に関して順に報告する. 前半 3 件の講演は文部科学省「次世代計算基盤に係る調査研究事業」に関する内容である. 後の 2 件は「富岳」における現在の取り組みに関する内容である.

「次世代計算基盤に係る調査研究」理化学研究所チーム アプリ**G**の取組 **2**

「次世代計算基盤に係る調査研究」
理化学研究所チーム システムソフト**G**の取組 **3**

「次世代計算基盤に係る調査研究」運用技術調査研究チームの取組 **4**

スーパーコンピュータ「富岳」における
システム省電力に向けた取り組み **5**

「**A64FX**システムアプリ性能検証**WG**」活動報告 **7**

# 「次世代計算基盤に係る調査研究」理化学研究所チーム アプリGの取組

岩下 武史(北海道大学), 高橋 大介(筑波大学)

1 件目は文部科学省「次世代計算基盤に係る調査研究事業」において，理化学研究所チーム（代表：近藤正章）のアプリグループが実施している活動についての講演である. アプリグループの主な活動は, アプリケーションの性質を規定する様々な指標の調査とアプリケーションの特性を反映した（マイクロ） ベンチマークの構築である. 一般的な指標に加えて, データ転送速度と計算速度の比である Bytes/FLOP や SIMD 化率などの指標を用いることで, アプリケーションの性質をより詳細に表現することができる. 調査の結果, 次世代計算基盤において成果を創出するために必要な技術要件が明らかとなり, アーキテクチャ, システムソフトウェアの要件調査や設計に寄与すると期待される.

アプリGは大きく純アプリGとCS Gの二つに大別され，両者が協調的に活動を行っている. これまでに, 純アプリ Gで は「富岳」などで行われている標準的な解析と大規模解析 (グランドチャレンジ) について調査が行われている. これらの解析における通信量, 演算量の調査により， 各Gにおいてそれぞれのアプリケーションの特性が明らかにされている. さらに,, 次世代計算基盤が本格的に稼働すると予想される2030 年頃において, サイエンスにおけるトップレベルの成果創出に必要な資源やプログラミングに関する要望についても調査が行われている. 今後は, アプリケーション間の連携を想定したワークフローベンチマークや, 通信低減アルゴリズムなど新アルゴリズムの導入による効果について検討がなされる.

続いて, CS G の報告がサブグループ毎に行われた. 機械学習サブ G では機械学習のベンチマークについてMLPerfからの選定を進めている. MLPerf では物体検出から自然言語処理など様々な分野を想定したベンチマークが用意されている. その中から, 分野や特徴が異なる 5 種類が選定された. ベンチマークサブ G では純アプリ G と連携してベンチマークを整備し, 一部はアーキ G に展開している. 科学技術計算サブ G は時空間タイリングや通信削減アルゴリズムなどの検討を行っている.

講演限定の内容として, 2030年頃に各アプリ分野が期待する成果についても述べられたが, 公開資料の内容と合わせて省略する. 最後に, アーキテクチャとアプリケーションは鶏と卵の関係にあり, 今後もアーキ G とシステムソフトウェア G と連携をとって進めていくと述べられた.

発表後 3 件の質問があった. 1 つ目はアーキテクチャ G からアプリ G に向けたアンケート内容についての質問があった. 回答としては, アンケートの目的はアーキテクチャを考える上で重要なアプリケーションのボトルネックに関する情報収集である. アンケートを通じてアプリケーションのボトルネックになりうるメモリアクセス・通信パターンといった内容がアーキテクチャグループに共有されたとのことだった.

2 つ目は今後のスケジュールに関する質問であった. 回答としては,「次世代計算基盤に係る調査研究事業」は複数のベンダ, 機関で分担されており, グループ内外の進捗を管理する必要がある. そのため, アプリケーション G だけでスケジュールを組めるわけではなく, アーキテクチャ G などとの間で密に連絡をとりながら調整しているとのことだった.

3 つ目は, サブグループも含め様々なアプリケーションがある中でどのアプリケーションに着目するかという内容であった. 2030 年のアーキテクチャを考えるにあたってキラーアプリが何かが

定まらないと進められないのではないかという観点での質問であった. 回答としては, アプリGの取りまとめの立場としては, 現時点で特定のアプリ分野を重要視することはグループの運営上考えておらず, あくまで各サブGの調査は横並びで進める方針であること, またキラーアプリの選定については, 調査結果に基づいて別途, 必要性に応じて決められるのではないかとのことだった. なお, 計算科学と機械学習の協調によるワークフロー型のアプリのように, 近年, 注目される新しい形のアプリケーションについては, より重要視されるのではないかとの補足があった.

## 「次世代計算基盤に係る調査研究」<br>理化学研究所チーム システムソフトGの取組

佐藤 賢斗(理化学研究所)

2 件目は 1 件目と同じ理化学研究所チームのシステムソフトグループが実施している活動についての講演である. システムソフトグループでは次世代計算基盤の具体的な性能・機能, 利用者のニーズを調査により明確化し, 必要なシステムソフトウェアの開発方針の検討を行う. 開発方針を検討する理由は, 2030 年に向けた限られた時間では全システムソフトウェアを国内のグループで開発し直すことはできず, 開発の優先度を決定し効率よく開発していくためである. 優先度の決め方としては, 移植性などの工数の観点, 商用・技術的な展開性, 海外の計算機システムセンタおよび利用者の動向なども踏まえて決定する.

続いて, 調査項目の詳細と今後の調査研究計画について述べられた. 調査項目は以下の3つである.

1. システムソフトウェアの動向調査
2. 国内で開発すべきか否かを検討するための調査
3. 類似ソフトウェアに関する差異の調査

調査1では, 既存および将来システムソフトウェアの移植性・生産性, 性能についての調査とアーキテクチャおよび計算機センターのシステムソフトの利用状況の違いについて調査を行う. 調査 2 では, システムソフトウェアの開発体制や移植性の観点から国内で開発すべきか分類を行う. 国内で開発すべき例としては富士通コンパイラや LLIO ソフトウェアがあげられ, 他のシステムソフトウェアは移植もしくはコミュニティと協力して開発する必要があるとしている. 調査 3 では, MPI ライブラリや数値計算ライブラリのような類似のシステムソフトウェアが複数存在する場合, それらの差異を調査する. 例えば数値計算ライブラリのうち, OneMKLやcuNumericは特定のプロセッサ向けに最適化されており, サポートする関数の種類にも違いがあると考えられる. このように各ライブラリが提供する開発環境の違いを調査することも, システムソフトウェアの選定に重要である. 今後の調査研究計画としては, 調査 1~3 を 2023 年は行い, 2023年末に最終報告書を提出する計画が示された.

発表後に 3 件の質問があった. 1 つめは, 優先度づけに用いる生産性をどのように定義しているかとの質問であった. 回答では, 一般的には変更が必要な行数で生産性の高低を議論するが, 実際には 1 行変更するのにかかる時間は異なる. そのため, 行数に加えて一行を書き換えるのにかかった時間なども考慮して生産性を定義していきたいとのことだった.

国内で維持するソフトウェアの例としてあげられた LLIO について, 過去と比べるとストレージの I/O 性能 が高速化しているのに, LLIO が必要なのかという質問があった. 回答では, LLIO を維

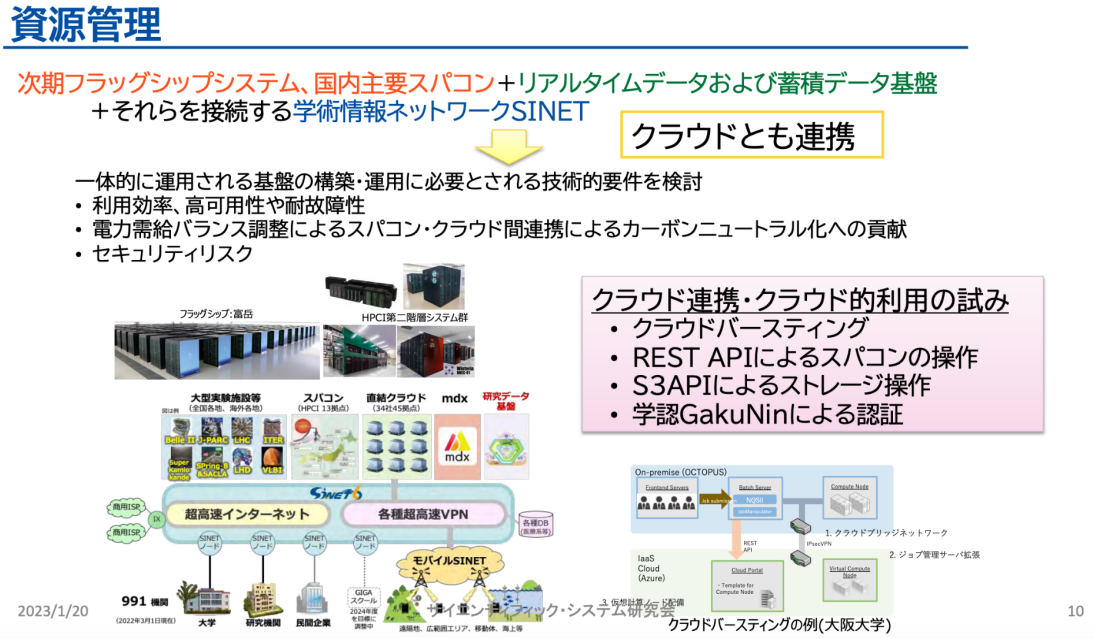
持することには固執しておらず，十分な I/O性能 が達成できていれば使わない可能性もあるとのことだった．

移行性に関する質問では，仮にアーキテクチャ G による調査結果でマイナーなアーキテクチャを採用する結論になった場合に，ソフトウェアの作り直しが多すぎるため システムソフトウェア G により却下されマイナーなアーキテクチャへの移行は実現されないのではないかとの意見があった．回答では，Intel の oneAPI のように多様なアーキテクチャで動作するフレームワークを用いて開発することで，アーキテクチャの変更には対応可能である．そのため，システムソフトウェアの都合でアーキテクチャの選択肢を狭めることはないとのことであった．

## 「次世代計算基盤に係る調査研究」運用技術調査研究チームの取組

塙 敏博(東京大学)

3 件目は「次世代計算基盤に係る調査研究事業」の運用技術調査研究チームの活動に関する講演であった．運用技術調査研究チームでは，計算基盤をより有機的に結合した，持続可能な次世代計算基盤の実現を目指している．次世代計算基盤では，計算機システム，データプラットフォーム，研究データ基盤，学術ネットワーク，クラウドなど研究者が望む資源を平易で柔軟かつシームレスに提供する．




最初に計算機システムの利用方法の将来像について述べられた．現状の計算機システムには，計算資源がオンデマンドに利用不可，資源利用量を予め宣言する必要性，システム間でのデータの移動が困難といった問題がある．そこで，研究データ管理基盤 GakuNin RDM をハブとして次世代計算機基盤を構築することで，これらの問題の解決を試みる．学認による SSO (Single Sign On) で認証を一元化し，各種データプラットフォームやクラウドストレージとの連携機能，GUIによる管理機能を追加することで次世代計算基盤において求められる資源提供が可能になる．

次は, SDGsの観点における次世代計算基盤の将来像が述べられた. 2030年度に温室効果ガスを2013年度から46%削減することを目指し, さらに, 50%の高みに向け挑戦することが閣議決定された. 一方, 計算機システムではプロセッサの消費電力および排熱が増加している. 温室効果ガスの削減目標を達成するには, 再生可能エネルギー由来の電力活用, 排熱再利用, 省電力化に向けた技術開発などの対策が必要になると予想される. しかし, これらの対策は計算機システムのシステム導入コストを増加させるため, 予算の都合上導入が困難という課題がある. 今後は排出権取引のように, 計算機システムの研究成果により削減された温室効果ガスなども, 計算機システムの排出権として計上できるような制度作りも一つの方向性として示された.

質問では,「京」の施設を再利用する都合上, 「京」から「富岳」への移行時に「京」を停止したが, 「富岳」共用開始までの間は「京」の利用者が使う計算機システムがない空白期間が発生した. 「富岳」から「富岳NEXT」への移行時にも同様の事態が起きるのではないかという意見があった. 回答では, 他の基盤センター等の導入計画次第なところもあるが, 「京」から「富岳」への移行時と同様の問題が起きないように考えていくことは重要とのことだった.

もう一つの質問では, 次世代計算機基盤を構築した際にボトルネックになるのはSINETで. 400Gbps に増強されたとはいえ, HPC ユーザが扱うデータのサイズを考えるとまだ不足しているのではないかとの意見であった. 回答としては mdx をモデルに複数拠点への拡張をすることで, ネットワークがボトルネックになることを回避できると考えられるのことだった.

## スーパーコンピュータ「富岳」における<br>システム省電力に向けた取り組み

篠原 誠（富士通株式会社)

後半の1件目は ”スーパーコンピュータ「富岳」におけるシステム省電力に向けた取り組み” という題で富士通株式会社の藤原誠による講演であった. 「富岳」は「京」の 100 倍のアプリケーション実効性能に加え, 高い電力効率を目標としている. 実際に「富岳」はプロトタイプでGREEN500の世界一を獲得し, 他にも史上初のHPC4冠獲得, 4期連続の世界一獲得と目覚ましい実績を残している. 本講演では, アプリケーションとシステムによるコデザインにより, 「富岳」がアプリケーション実行性能を低下させずに省電力化を実現した方法について述べられた.

最初は「富岳」に搭載されている A64FX に関する省電力化の取り組みの紹介が行われた. A64FX には SIMD におけるデータ読み出しの電力に着目し, 不用意なデータ読み出しを減らす仕組みが搭載されている. 具体的には L1 Cache における不要な way の読み出しを防ぐ Long Latency L1 Cache Access と同じデータを何度も呼び出すことを防ぐ Operand Register Bypass の 2 つの機能が搭載されている. これらの機能により電力性能目標 15GFlops/W 超えの, 16.88 GFlops/W の性能を達成した. その他にも, A64FX には省電力用の 2 つの機能が搭載されている. 1 つ目は周波数, 演算ユニット数などを変更するパワーノブである. 2 つ目はアイドル状態の CPU コアを停止するリテンションである. 省電力機能を様々なアプリケーションに適用したところ, スレッド数などアプリケーションの特徴によっては性能が低下せずに省電力化できるケースが確認できた.

## A64FXの電力制御：パワーノブ

FUJITSU

**性能を下げずに余分な電力を削減するため、パワーノブを採用**

アプリケーション特性に応じて、使用していない部分を停止し、電力効率を向上

| パワーノブ | 概要 | 設定可能な値 |
|---|---|---|
| **周波数変更** | **コア全体のCPU周波数を制御** | **・2.2GHz（ブースト）<br>・2.0GHz（ノーマル）<br>・1.6GHz** |
| **浮動小数点演算ユニット制御** | **使用可能な浮動小数点演算器の数を制御<br>エコモード(※)の制御<br>※電力安定化のための電力消費を抑えるモード** | **・FLA/FLB、エコモード無効**<br>・FLA Only、エコモード無効<br>**・FLA Only、エコモード有効** |
| **固定小数点演算ユニット数** | 使用可能な固定小数点演算器の数を制御 | ・EXA/EXB<br>・EXA Only |
| **命令発行数** | コアが同時に処理する命令数を制御 | ・4命令<br>・2命令 |
| **メモリスロットリング** | メモリアクセスコントローラーとメモリ間のバス使用率を制御 | ・バス使用率 100%～10%<br>10%単位 |

14

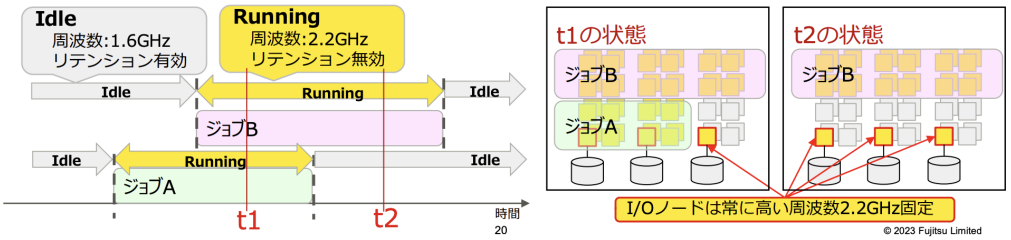
© 2023 Fujitsu Limited

次に，先述した A64FX のパワーノブとリテンションを用いた省電力化の取り組みの紹介が行われた．A64FX のパワーノブとリテンションは Power API として提供されており，システム管理者とエンドユー ザの両方が利用できる．システム管理者による利用例として，ジョブ運用ソフトウェアにおける電力制御が紹介された．1 つめはジョブ実行に合わせた省電力モード切替で，コアへのジョブの配置の有無で周波数設定を変えている．2 つめはノード種別を意識した電力制御で，ファイル I/O を担当する I/O ノードの周波数を高く固定する．これにより，省電力化によるファイル I/O 性能低下を防止する．

## ジョブ運用ソフトウェアによる活用

FUJITSU

- **ジョブ実行に合わせた省電力モード切替**
  - ジョブの開始・終了に応じてモード（Idle/Running）を自動的に切り替え
  - 運用に合わせて、システム管理者はそれぞれのモードの設定を定義可能
- **ノード種別を意識した電力制御**
  - 計算ノードはディスクレスのため、ファイルI/OはI/Oノードを経由
  - I/OノードにファイルI/O性能影響を考慮した設定が可能（例：2.2GHz固定）




20

© 2023 Fujitsu Limited

最後に，運用に関する省電力化の取り組みが紹介された．「富岳」のシステム消費電力を超えないように制御することは，安定稼働に不可欠である．そのために，BoB（Bunch of Blades）単位での段階的な電力キャッピングを実行している．第一段階では，CPU の周波数を下げ，第二段階では計算ノードの強制停止を行う．しかし，実際に電力キャッピングを適用してみると不用意な電力キャッピングが発生した．その原因はノード毎の消費電力の個体差にあり，ノードを交換することでラック毎の電力消費量を平均化し対処した．

当日はノード毎の消費電力の個体差の話についての質問があり, ノード電力の個体差をあらかじめ計測し, ラック毎の消費電力が同じになるように配置すれば未然に防げたのではないかという旨の内容であった. 回答では, ノード数16万と大量のノードを扱っている都合上, ノードを作っては段階的にシステムに投入していた. そのため, 全ノードをセットアップしラック毎の電力を計算してから導入するというプロセスは取れず, 未然に防ぐことは不可能だったとの内容だった.

## 「A64FXシステムアプリ性能検証WG」活動報告

片桐 孝洋 (名古屋大学)

最後の講演は, A64FXシステムアプリ性能検証WGによる成果報告であった. 当グループの活動目的は, コンパイラなどのシステムソフトウェアと協調した性能最適化を行うためのノウハウの集約と共有である. A64FXシステムアプリ性能検証WGの活動期間は2020年11月~2022年10月で, 計10回の会合が開かれた. 前半 3 ~ 4 回では推進委員が保有するアプリケーションについて現状の問題点を抽出した. 中盤 3 ~ 4 回では富士通が提供するコンパイラの問題点・課題点について評価検討を行った. 後半 2 回では 2022 年富士通SS研科学技術計算分科会にて成果報告を行った. 以降は, 当グループで行った問題点の抽出からコンパイラでの解決方法までの報告が各事例毎に行われた. 事例の量が膨大であるため, 本レポートでは富士通コンパイラの改善に直接的につながったと考えられる事例のみを抽出して紹介する.

1件目の事例は, JAXA の乱流燃焼コード LS-FLOW HO (LSHO) である. 性能低下の原因は配列代入の統合によるレジスタ不足である. 解決方法として並列リージョン拡張を無効にするオプションの利用が富士通から提示された. 今回の事例では配列代入を一つのループに統合しようとした結果, false sharing によりキャッシュが活用されずディスクスピルが発生していた. オプションを付与することで並列リージョン拡張機能が抑制され, 該当の統合処理が行われなくなる. また, 並列リージョン拡張機能が動作したことをログとして表示していないことが, 原因発見の遅れにもつながるという点で問題があるという知見が得られた.

2 件目の事例は CFD プログラムの PRIMEHPC FX1000 向け高速化チューニングである. FX1000 のレジスタの小ささがボトルネックになると考え, !ocl loop_fission を用いた自動ループ分割を用いることで, ディスクスピルの防止を試みた. しかし, 結果からは L1 キャッシュミスが発生しており, ループの分割数を決定する !ocl loop_fission_threshold(N) にチューニングの余地がありそうとの報告であった. 報告を受けて富士通側で自動ループ分割に関する更なる検証が行われた. その結果, ループ分割数をデフォルトより少なくした !ocl loop fission threshold(90) が最速であることが判明した. また, L1 キャッシュミスを減らす指示行を入れたところ, L1 キャッシュミスは減ったが命令数の増加のオーバヘッドが上回り速くならなかった. 結果からは, L1 キャッシュミスで遅くなっているわけではないことが判明した.

3 件目の事例は GF-based NN predictor である. 富士通コンパイラによる最適化を試みたところ6重ループの箇所に対して, 富士通コンパイラは最内ループを SIMD 化しており速くならなかったとの報告であった. 富士通側の回答としては, 最内ループを SIMD 化するのは仕様であり外側ループの SIMD 化は大規模な新規開発を要するため計画していないとのことだった. 対策として !ocl clone を追加し, 一定のループ数以下の場合は別ループとすることで高速化できると報告さ

れた. しかし, !ocl clone と OpenMPI の組み合わせが動作しないという新たな課題も発見されるなど, 本事例は複数の課題の発見に繋がった.

終了後, 2 件目の事例に携わっていた方から, CPUの性能解析ツールをL1, L2キャッシュに関する性能をより詳細に確認できるよう整備してほしいとの意見が出た.