1. 調査研究の概要

1.1 調査研究の背景と目的

次世代の計算基盤には、従来のサイエンスにとどまらず、広範なSDGs・Society 5.0の実現に向けた課題解決のためのプラットフォームとしての役割が一層求められる。特に、今後の科学では、いわゆる「研究DX」により新たなイノベーションを起こすべく非常に高度なデジタルツインを中心としたものとなり、そのためには現象をモデル化することによる演繹的シミュレーションと、学習・AIに基づく現象の帰納的シミュレーション、さらには大規模なデータを収集・処理基盤と、データの両方のシミュレーションとの同化、などが密に連携する計算基盤の構築、及びその開発・運用が鍵となる。このような次世代計算基盤では、大規模高速な計算・データ処理が行えるスーパーコンピュータがその中心的な役割を担う。特に、比較的狭いシミュレーション領域を対象とした過去のものや、AIや特定領域の計算に特化したマシンと異なり、デジタルツインの基盤としてのスーパーコンピュータは、広範な計算手法・シミュレーション技法や大規模データを駆使しつつ、それらが密に連携しながら全体のワークフローの実行が可能である必要があり、単に特定のベンチやアプリで高い性能を達成するのみならず、幅広いアプリケーション分野で高効率を達成できる計算基盤として特質が要となる。

しかしながら、その実現は非常に困難を極めると予想される。一般に、高い性能と広い適用可能性は相反する要求でもあり、現在理化学研究所計算科学研究センターで運用されているスーパーコンピュータ富岳の開発では、ハードウェアとアプリケーションのコデザインをHPCの産学オールジャパン体制で推し進めることで、それを実現可能なものとした。一方、次世代計算基盤では、ムーアの法則の終焉が近づきつつあるなか、半導体プロセス技術の進歩による演算あたりの大幅な電力改善が見込めず、高い性能をあまねく達成することはさらに難しくなると予測されている。実際、計算機科学コミュニティが調査・執筆したNGACI白書でも、2028年度に実現可能なシステムでは、許容電力の大幅増を仮定しても、富岳の3.3倍から10倍程度の性能向上にとどまると予想されている。また、近年のソフトウェア開発における複雑性の増大も、高い性能と広い適用可能性への大きな足かせになる。従来型の浮動小数点演算ピーク性能重視のシステム設計では、一部のベンチマークでは一見華々しい性能を見せつつも、実際のアプリケーションとの性能乖離は絶望的となり、今後立ち行かなくなることは明らかであり、またソフトウェアのエコシステムの重要さやコストを軽視した計算基盤では、開発しても使うことのできないシステムになるであり、研究DXの基盤としては意味のないものとなる。

性能面に関しては、様々な文献や我々のベンチマークや性能モデルの多岐にわたる研究調査でも、大多数のアプリケーションはメモリやネットワークバンド幅など、基本的にデータ移動の性能によって律速される特徴を持つことがわかっている。次世代計算基盤では、FLOPSで語られるピーク性能重視から脱却し、データ移動、すなわちByteにより表現される指標をアーキテクチャとアルゴリズムの点から最適化・高効率化していくことが重要となる。本調査研究では、上記理念の達成に向け、アーキテクチャ設計の基本理念として演算精度も考慮しながら必要な計算性能は確保しつつ、電力制約の下でデータ移動を高度化・効率化する"FLOPS to Byte"指向のシステム構築を、アーキテクチャ開発からアルゴリズム設計、アプリケーション技術に至るまで実践し、実効的な性能を向上させるための次世代計算基盤を調査研究することが目的である。

アーキテクチャの調査研究では、電力制約の下でデータ移動を高度化・効率するデバイス技術やアーキテクチャ技術を中心に調査研究を実施する。近年までの大規模スーパーコンピュータでは、レイテンシよりも演算とデータ転送スループットを重視し、かつ並列度を上げた際の実効効率低下を緩和させるために、弱スケーリング思想に基づく設計が重視されてきた。一方、機械学習における推論処理や創薬系のアプリケーションでは同一サイズの問題を高速に解く強スケーリングが求められ、半導体技術によるDRAM容量の増加傾向も鈍化しつつある。これら背景を踏まえ、アーキテクチャ検討項目として、特に3次元積層メモリ技術の活用、データフロー的観点も踏まえた強スケーリング向けの要素技術、チップ間直接光通信技術を特に意識しながら、関連するアーキテクチャ技術を国内外複数ベンダと共に調査研究を実施する。2023年度は、特に次世代計算基盤に向けたアーキテクチャの絞り込みのための調査研究を行った。

システムソフトウェア・ライブラリの調査研究に関しては、これまでのフラッグシップや第二階層システムのソフトウェアの開

発工数や利用状況、そして既に構築されているエコシステムとの親和性も踏まえ、ソフトウェア資産として何をベースに、国内でどこまでを開発すべきか、もしくは国際協調が必要なソフトウェアを優先度つけて明らかしつつ、今後のロードマップを策定することが必要である。また、次世代計算基盤を産業界も含む幅広い応用分野での活用を促すためにも、従来のシステムソフトウェアだけでなくデータ利活用の促進、機械学習技術と第一原理シミュレーション、さらには大規模リアルタイムデータ処理の高度な融合、従来の共用HPCシステムとは次元の異なる高セキュリティの担保、などを主要検討項目とし、次世代計算基盤として行うべきソフトウェア開発について調査研究を行う。2023年度は、システムソフトウェア開発戦略検討や、類似ソフトウェアに関する調査研究を実施した。

アプリケーションの調査研究に関しては、従来のサイエンスをさらに進化・深化させるのみにとどまらず、社会科学やSociety 5.0といった新しい応用分野への展開も見据え、アーキテクチャとアルゴリズムとアプリケーションの三者が連携するコデザインに基づく次世代計算基盤構築に向けた調査研究を実施する。特に、複数アーキテクチャを統一的に評価するための広範なベンチマークセットを構築し、それを利用したアーキテクチャ評価結果を踏まえてアルゴリズムやパラメータの改善を検討する。また、それをベンチマークセットへと更新し、性能モデルを構築した上で、探索的な評価を行う。このサイクルを回した新たなコデザインにより、各アプリケーションで高い実効性能を得るためのアーキテクチャとアルゴリズムを探求する。これにより、今後アーキテクチャの進化に伴って、どのようなアルゴリズムのクラスが大幅な進化が見込まれるか、という指標も抽出し、今後の発展につなげる。2023年度は、アーキテクチャ絞り込みのためのシステム評価も可能とするベンチマークセット更新や、ターゲットサイエンスのアップデートを中心に調査研究を実施する。また、新しいアプリケーション分野の開拓に向けた有識者からのヒアリングも行った。

フラッグシップシステムの開発にあたっては、日本の半導体戦略とも整合していくことが重要であり、特に今後の日本が持つべき半導体技術の強みを活用しつつさらにそれを伸ばすための戦略や、開発されたシステムの幅広い産業展開を見据えた調査研究へのフィードバックも実施する。これらを総合し、ポスト「富岳」時代の次世代計算基盤の具体的な性能や機能等について検討を行うことが目的である。

1.2 調査研究の体制

上記の調査研究を実施するために、国内外の主用なHPC関連のベンダと、HPCI第二階層を構成する国内のスーパーコンピューティングセンタ、また様々なアプリケーション分野をリードする国立研究所の研究者と共に、以下の体制で調査研究を実施した。

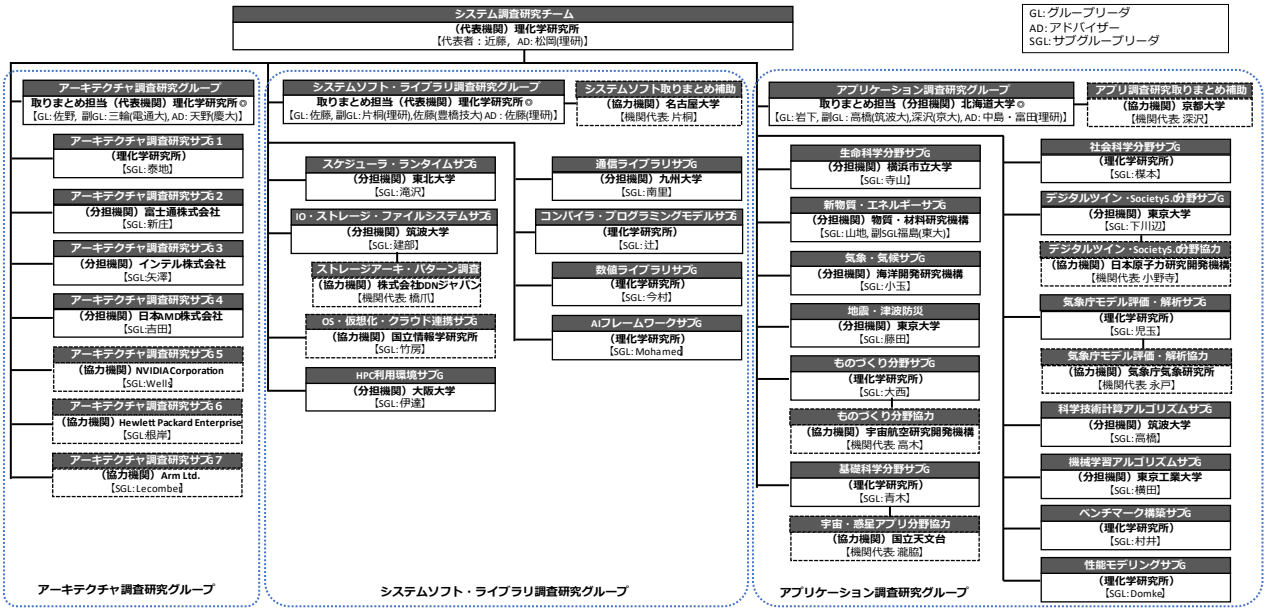


図 2.1.1.1 調査研究の体制

2. アーキテクチャ研究グループ

2.1調査研究の概要および方針

2.1.1 目的と方法

アーキテクチャ研究グループでは、科学技術研究分野、産業分野、および社会からの利用ニーズに対して有望なシステムの選択肢の探索およびその実現可能性の提示を目的とし、ムーア則の終焉や半導体技術およびパッケージング技術等の発展を見据えつつ、システム全体やその構成要素について考え得る技術的可能性に関する調査検討を行う。そのために、例えば、1. 汎用プロセッサ（CPU）、2. DRAM、3. SRAM（キャッシュメモリ等）やその3次元積層技術、4. アクセラレータ、5. 相互接続網、6. ストレージ、7. CMOS技術やパッケージング技術（3次元積層、チップレット等）、8. シリコンフォトニクス、9. 信頼性・可用性・保守性、などについて、技術動向と共に次世代計算基盤システム開発に適した将来技術を調査・探索し、その有効性および実現可能性を評価する。

ハードウェアシステムの開発および製造・設置に関しても具体的な検討を行うために、アカデミアに加え、関連分野において長年に渡り技術開発とシステム販売に携わってきたベンダを分担機関あるいは協力機関とし、上記の調査検討を実施する。特に、昨今の半導体業界やITC業界では国境を跨ぐ世界的なサプライチェーンや技術連携が不可欠であり、次世代計算基盤開発においても日本単独で全ての技術を網羅することは難しいと予想されることから、ベンダとして国内外の企業に参画頂き、特に日本が独自に保有し発展させるべき技術と国際的に連携すべき技術の候補を明らかにしつつ、開発すべきシステムとそのアーキテクチャの選択肢を提示することを目指す。半導体技術等の原理的限界と、技術的に突破できる見込みのあるそれ以外の課題を精査し、次世代計算基盤に求められる要件に対して優先順位を設定した上で解決策の実現可能性を検討する。

本年度では、グループとりまとめA0とサブグループA1～6の体制に加え、新たなサブグループであるA7を追加で設置し、上記の調査研究を実施した。

A0. アーキテクチャグループとりまとめ	担当機関：理化学研究所（代表機関）
A1. アーキテクチャ調査研究サブグループ1	担当機関：理化学研究所（代表機関）
A2. アーキテクチャ調査研究サブグループ2	担当機関：富士通株式会社（分担機関）
A3. アーキテクチャ調査研究サブグループ3	担当機関：インテル株式会社（分担機関）
A4. アーキテクチャ調査研究サブグループ4	担当機関：日本ＡＭＤ株式会社（分担機関）
A5. アーキテクチャ調査研究サブグループ5	担当機関：NVIDIA Corporation（協力機関）
A6. アーキテクチャ調査研究サブグループ6	担当機関：Hewlett Packard Enterprise（協力機関）
A7. アーキテクチャ調査研究サブグループ7	担当機関：ARM Ltd.（協力機関）

「A0. アーキテクチャグループとりまとめ」では、以下の項目について調査研究を実施した。詳細は、それぞれの節に記載する。

- 昨年度に続く、半導体技術・パッケージング技術・アーキテクチャの技術動向の調査（2.1.2 節）
- 昨年度のものに追加したベンチマークプログラムの特徴解析によるワークロード分析（2.1.3 節）
- 昨年度に得られた複数のアーキテクチャ候補の詳細化と絞り込みに向けた評価（2.1.4 節）

アーキテクチャ調査研究サブグループ1～7では、マイクロアーキテクチャ、ノードアーキテクチャ、システムアーキテクチャ、および関連するシステムソフトウェアやベンチマークによるアプリケーションの性能推定等の個別に設定する調査研究対象について、技術的な調査検討を行った。詳細は、2.2～2.8節について述べる。

2.1.2 技術動向調査

2.1.2.1 技術動向調査の概要

スーパーコンピュータ、特にフラッグシップ機は、高い演算性能と電力性能を実現するために最先端のハードウェア技術を駆使してシステム構築が行われる。そのため、次世代計算基盤のアーキテクチャ検討においては、システムを構成する様々なハードウェアの技術動向を十分理解した上で、2028年頃までに実用化可能な技術を見極める必要がある。

そこで本研究グループでは、昨年度に続き調査を行い、種々のハードウェア技術動向を更新した。動向調査の対象とした技術は、具体的には、半導体、パッケージング、シリコンフォトニクス、汎用プロセッサ、メモリ、アクセラレータ、ノードアーキテクチャ、CPU接続、ネットワークである。以下では各技術の動向調査結果を報告する。

2.1.2.2 半導体・パッケージング・シリコンフォトニクス

昨年度の報告書では、以下の調査結果が記されている。半導体の微細化は2030年でも継続する見通しである。IRDSテクノロジロードマップによると、1 nm eq プロセスが2031年に実現される見通しであり、5 nmプロセスから1 nmの微細化で面積あたり4倍以上の実装密度を実現する。トランジスタ構造は3 nm近辺を境に、FinFETからLGAAに変化する。ウェハーコストは微細化とともに上昇する傾向があるが、ダイ面積も微細化とともに縮小するため、同一機能を有するダイの製造コストは微細化により低下する。ウェハ上の欠陥密度はプロセスによらず概ね一定の傾向（TSMC N7プロセスで0.09 欠陥/cm^2）があるため、同一機能を持つダイを製造する場合、チップ面積に比例して歩留まりは改善する傾向にある。以上より、微細化による性能改善・価格低下・歩留まり改善は依然として進展する。

一方で、計算重視の計算基盤を持続的に成長させる上で、微細化の鈍化が今後の大きな懸念事項である。さらに前述の通り、ウェハ上の欠陥密度はプロセスによらず概ね一定の傾向があるため、ダイ面積を拡大して並列ユニット数を増やすと歩留まりが低下し、逆に製造コストの増大に繋がる問題点がある。このため、ダイを複数枚に分割して、2次元方向に接続するチップレット技術や3次元集積技術が今後の技術トレンドとして研究されている。NVIDIA社GPUのチップレット技術トレンドを見ると、チップを接続するインターポーザのサイズが3年ほどでおよそ倍増のペースで成長しており、演算ユニットに搭載するトランジスタ数は直近5年で20倍、メモリ搭載数が8倍に進化している。複数のチップレットを相互接続するための規格としてUniversal Chiplet Interconnect Express（UCIe）が2022年頃から急速に策定されはじめており、機能をわけたチップレットの相互接続を目指して今後の持続的発展が注目されている。従来のチップ大面積化による機能改善と代わって、省面積チップレットの3次元統合が設計コストを下げながら性能改善を実現する有力手段として考えられる。

3次元実装方式として、積層されたチップの電気接続をThrough Silicon Via (TSV)により行うことが現在主流であり、メモリや演算ユニットの実装密度を高める。メモリチップを3次元方向にTSV接続したHBM (High Bandwidth Memory)は近年では10層以上積層し、その容量は10 GBを超えている。この3次元積層されたチップを水平方向（2次元方向）に密に並べるために、TSVやインターポーザを排除したEmbedded Multi-die Interconnect Bridge (EMIB) がIntel社により開発されており、2次元方向への高密度実装の開発も活発に行われている。

データセンタなど、データ転送がボトルネックとなるアプリケーションに対して、スイッチング方式の最適化などによりスループット帯域の改善は行われている一方で、サーバやラック間のデータ転送を行うモジュールの速度が追いついていない。1つのブレークスルーとしてシリコンフォトニクス技術が重点的に研究されている。電気配線を用いた通信では配線の寄生成分によりバンド幅が狭くなり、電力のロスが発生する一方で、光通信はこのようなバンド幅の低下や電力ロスが深刻ではないためである（現行の光モジュールの性能については、2.1.2.5を参照）。光通信モジュールと、ASIC間は現状では電極を介してプラグ接続されているため、集積度やバンド幅、電力効

率が律速される問題点がある。先述したチップレットの相互接続技術の急速な発展を背景に、Co-packaged Optics (CPO)が重点的に開発されている。例えば光変調器を搭載したプロセスで製造し、ドライバや制御回路をCMOSチップレットで混載設計することで、基板やパッケージ間を繋ぐ配線や減衰信号補償用のDSPが不要になる。しかし、依然としてデータセンタに必要なスループット帯域に追いつく性能には至っておらず、1つの問題点として集積度がCMOS回路より悪い点が挙げられる。チップレット積層技術や光素子そのものの高集積化が重要課題である。

昨年度に続く調査の結果、以上の技術動向には以下の更新が見られた。半導体の高密度実装構造について、本年度は1nmノード以降の微細プロセスのトレンドとして、トランジスタ構造の3次元化が見られた。具体的にはIntelやTSMCがnMOSFETとpMOSFETを上下方向に配置し、最大50%の面積削減を実現するComplementary Field-Effect Transistor (CFET)技術を公表した。TSMCは48nmのゲートピッチ、Intelは60nmのゲートピッチの実現に成功している。IRDS 2022によると、トランジスタをFinFET (3nmノード、2022年)からCFET (1.5nm、2031年)にすることで、デジタル回路ブロックの面積がおよそ55%に、さらに3次元集積を施すことで2037年には0.5nmノードにて8%の面積になることが予想された。

Hybrid Bondingによる3次元チップ積層方式として、AMDはGPUアクセラレータMI300を発表し、2.5DのHBM実装と比べておよそ2.5倍の電力効率を実現している。IBMによるAIアクセラレータNorthPoleはNvidia社のL4と比べて約2.5倍のエネルギー効率改善を実現している。また、異なる高密度実装手法のテクノロジトレンドとして、チップ背面から電源を供給するPowerViaをIntelが提案し、90%を超えるセルユーティライゼーション、6%の動作性能改善を実現している。2024年よりIntel A20世代から実装予定である。SK HynixがDRAMのHybrid bondingに成功し、従来のマイクロバンプ方式のHBMとの性能差別化の可能性を示した。

CPOについても研究開発が進んでおり、TSMCはチップレットベースのCPOの性能を見積もっている。従来のIIIV族プラガブルベースの光モジュール（1.6Tbps）からシリコンフォトニクスベースのCPOに変更することで、約5倍の6.4Tbpsのバンド幅改善を見積もった。

2.1.2.3　汎用プロセッサ・メモリ

- **汎用プロセッサ**：以下では汎用プロセッサの技術動向を述べる。

  昨年度の報告書では、以下の調査結果が記されている。HPC用途では主にIntel、AMD、IBM、富士通の各ベンダが汎用プロセッサを提供しており、世界トップレベルの性能を有するスーパーコンピュータは上記のいずれかのベンダのプロセッサを採用することが多い。HPC向けに設計された汎用プロセッサでは多数のコアと各コアに搭載したSIMD演算ユニットによってピーク演算性能を稼ぐアーキテクチャを採用する傾向にあり、現在は数十個のコアとコアあたり1～2個の512ビットSIMD演算ユニットを搭載したものが主流となっている。また、Intel第4世代Xeonスケーラブルプロセッサ やIBM Power10等の一部のプロセッサでは、AI処理で多用される行列演算を加速するために行列演算ユニットの搭載を開始している。上記の傾向は今後も続くと予想されることから、ソケットあたりのコア数、SIMD演算ユニットの幅と数、行列演算ユニットの幅と数が2028年頃にどの程度まで増大するかは注意深く見守る必要がある。なお、2023年4月現在、HPC向け汎用プロセッサのピーク演算性能は最大で7.37TFLOPS（AMD EPYC 9654Pにおける倍精度浮動小数点演算性能）に達しており、TDPは350W前後、電力性能は20GFLOPS/W程度である。メインメモリに関してはこれまでDDRが主体であったが、近年は富士通A64FXやIntel Xeon-MaxなどのHBMをサポートする汎用プロセッサも登場し始めている。HBMをサポートする汎用プロセッサはバンド幅律速のアプリケーションに対してGPUとほとんど変わらない性能を示すと期待されることから、このようなプロセッサの技術動向は今後注視していく必要がある。また、拡張インターフェイスとしてPCIe 5.0を数十から百数十レーン搭載したものが主流となっている（IBM Power9で採用されたNVLinkはPower10では廃止されている）。

半導体製造技術とアーキテクチャの進歩によって汎用プロセッサの性能は大きく向上しているものの、単純にピーク演算性能や電力性能で比較すると、汎用プロセッサはGPUを始めとするアクセラレータとは依然として10倍以上の開きがあるのが現実である。汎用プロセッサがアクセラレータよりも電力性能で劣るのは、命令をアウトオブオーダ実行するための様々なユニットが汎用プロセッサには必要であり、演算以外の部分で消費される電力が大きいためである。これはプロセッサが汎用であり続けるための本質的な性質であり、汎用プロセッサとアクセラレータの電力性能比は2028年頃においても現在と同程度であることが次世代先端的計算基盤に関する白書でも予想されている[1]。また、近年のスーパーコンピュータ開発においてはシステムに課された電力バジェットが大きな制約となっており、この制約は今後厳しくなることはあっても緩くなる可能性は低い。したがって、2028年頃に高いピーク演算性能を有するシステムを実現する上では、何らかのアクセラレータの採用は避けて通ることができないものと思われる。その一方で、アクセラレータによって加速可能なアプリケーションは一般に一部のドメインに限られることから、アクセラレータを採用する場合は当該アクセラレータが加速可能なアプリケーションのドメインを慎重に検討するとともに、当該アクセラレータでは加速できないアプリケーションの加速方法についても考える必要がある。

昨年度に続く調査の結果、以上の技術動向には以下の更新が見られた。汎用プロセッサに関しては、本年度に入り、IntelとAMDはそれぞれ第5世代Xeonスケーラブルプロセッサと EPYC 97x4プロセッサの販売を開始した。その結果、汎用プロセッサのピーク演算性能は9.22TFLOPS（AMD EPYC9754における倍精度浮動小数点演算性能）、電力性能は25.6GFLOPS/Wにまで向上したものの、性能面でアクセラレータが優位な状況に変わりはない。例えばAMD MI300xのピーク演算性能は倍精度で163.4TFLOPS（EPYC 9754の18倍）、電力性能は218GFLOPS/W（EPYC 9754の8.5倍）である。また行列演算ユニットに関しては、第5世代Xeonスケーラブルプロセッサでは16x16x32の行列積を16サイクルで実行するようになった[2]。Intelの行列演算ユニットが現在サポートするデータ型はBF16とINT8のみであるが、Granite Rapidsでは半精度複素数型のサポートも予定されている[3]。

- **メモリ**：ここでは、主にオンチップおよびオフチップのメモリ技術の動向調査について述べる。

昨年度の報告書では、以下の調査結果が記されている。オンチップメモリとして主要なメモリ素子としては SRAM が用いられている。これまでは、SRAMは基本的にプロセスのスケーリングに伴って微細化され、高集積化と低消費電力化を実現してきた。しかし、10nm 以降のテクノロジではロジックの微細化は鈍化しつつも進んでいるが、配線に関しては微細化されると抵抗が増加することもあり、ロジックとの乖離が広がっている。キャッシュなどに用いられる大容量 SRAM は配線領域が支配的となるため、大容量化が難しくなってきている。また、電圧についても安定動作のためには現在の電圧が限界となり、これ以上の低電力化は難しくなってきている。一方で、ハイブリッドボンディングやTSV（スルーシリコンビア）による3次元実装技術により、SRAM 層をロジック層の上に搭載することにより、高バンド幅や大容量化を実現する試みも一部で実用化されており、3 次元実装技術の進展がオンチップメモリでも重要となってきている。また、3次元実装技術では、DRAM をオンチップに搭載することも可能となるため、大容量化の方向としては検討項目となる。ただし、アクセス速度は SRAM に比べると遅いため、階層化は必須となる。また、DRAM ではリフレッシュが必要となるなど課題も多いため、これらを解決する不揮発型メモリの開発も行われており、実用化に向けた動向について注意すべきである。

オフチップメモリとして最も使われているのは DRAM で構成されるデバイスである。DRAM においては、第 5 世代の規格である DDR5 が 2021 年末から発売が開始され、最新の CPU では利用され始めている。DDR5 は、チップモジュールあたり 16Gbit の容量を有し、1.1V 供給電圧の下で、現在出荷されている DDR5-4800 では 38.4GB/s であるが規格的には 51.2GB/s までを視野に入れている。次世代規格として DDR6 が出てくるのは 2027 年頃と見られ、その動向に注意が必要である。一方、より高いメモリバン

ド幅を要求する HPC 向けのプロセッサ（富士通 A64FX や Intel Xeon-Max など）やアクセラレータ（NVIDIA H100 や NEC SX-Aurora TSUBASA など）には TSV とシリコンインターポーザを用いた HBM が用いられている。HBM は DDR と比べて高いバンド幅と電力効率を有し、最新の HBM2E では 410GB/s のメモリバンド幅を実現している。次世代の HBM3 の仕様書も 2022 年初めに JES238 として公開され、ピンあたりのデータ転送速度を 2 倍に相当する 6.4Gbps に拡張し、デバイス当たり 819GB/s とした。TSV スタックも最大 12 とし、将来的には 16 への拡張性も確保している。層あたりの容量密度も8Gb から 32Gb とし、デバイス当たり 64GB までサポートできる。ただし、初期の製品は 16Gb となる見込みである。一方で、これらの二つのメモリデバイスには、容量、メモリバンド幅、コストのトレードオフがある。このため今後は、これらのトレードオフを考慮しながら、DDR と HBM が引き続き高性能計算システムのメモリ素子として利用されていくことが予想される。さらに、新しい不揮発メモリデバイスについても、開発が進められており、電力・コスト・容量などの観点から動向を注目する必要がある。

昨年度に続く調査の結果、メモリ各社からHBM3Eの量産が発表された。データ転送速度は9.2-9.8Gb/s、デバイス当たりで1.2-1.28TB/sに拡張されている。2024年第2四半期では24GB版の出荷が予定されており、36GB版の開発・評価も進んでいる。HBM4についても規格が検討されており、2025年頃に規格が出てくる見込みである。一方、DRAMの３D実装については、研究は各所で行われているが、実用化に向けた期待が高まっている。例えば、2023年のISSCCのPlenaryで発表したAMDのLisa Suの講演[4]では、すでに出荷が行われているSRAMを3D実装した3D V-Cacheの他に、DRAMのスタッキングを将来の技術と紹介しており、実用化に向けた開発については、今後も注目する必要がある。

[1] NGACI、White Paper on Next-Generation Advanced Computing Infrastructure、ver. 1.0.0 (2020)

[2] Intel Corp., Ushering in a New Era of Accelerated AI on Intel CPUs, https://www.intel.com/content/www/us/en/developer/articles/technical/usher-in-a-new-era-of-accelerated-ai-on-cpus.html

[3] Intel Corp., Intel Architecture Instruction Set Extensions and Future Features, revision 50, September 2023.

[4] ISSCC 2023 Plenary – Lisa Su: Innovation For the Next Decade of Computer Efficiency, ISCC Videos, https://www.youtube.com/@ISSCCVideos.

2.1.2.4　アクセラレータ・ノードアーキテクチャ・CPU 接続

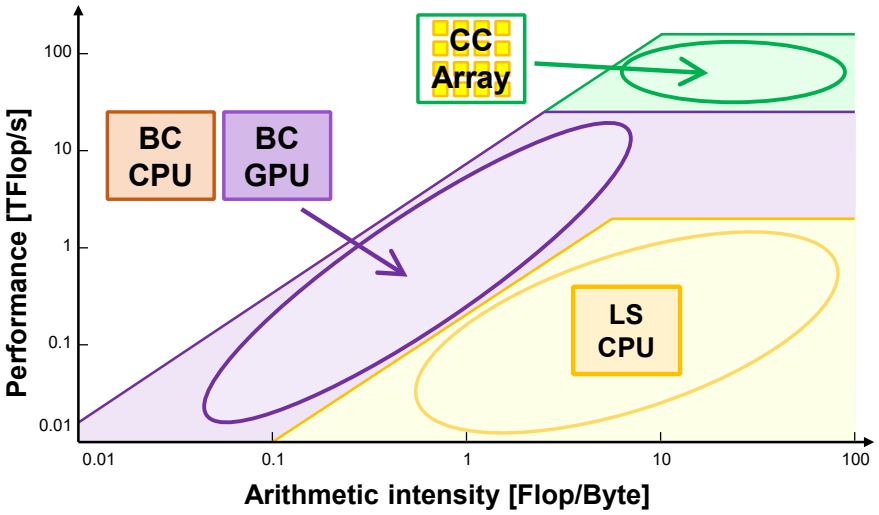


図 2.1.2.1　Roofline モデルとノードを構成する基本アーキテクチャ

昨年度の報告書では、以下の調査結果が記されている。次世代先端的計算基盤に関する白書（NGACI, https://sites.google.com/view/ngaci/home）にもまとめられているように、高性能計算機を構成可能な基本アーキテクチャ要素としては、

- Latency Sensitive CPU [LS]
- Bandwidth Centric CPU または GPU [BC]
- Compute Centric Accelerator [CC]

の3つが考えられる。図 2.1.2.1は、Rooflineモデル上においてこれらの基本アーキテクチャの達成可能性能領域を示したものである。横軸は演算強度[Flop/Byte]で、縦軸は性能[TFlop/s]を示す。LSはデータセンタ等で用いられるようなハイエンドCPUのことであり、スレッド処理の遅延を短縮して様々なワークロードに対する性能を高めるような汎用プロセッサである。その特性、および大容量の主メモリを搭載するために、DDRメモリが採用されていることが多い。また、遅延を削減する様々な機構のためにピーク性能や電力性能比はその他の基本アーキテクチャよりも低い範囲となる。

BCは、HBMメモリを搭載した計算処理のスループットを指向したハイエンドCPUやGPUであり、例えば、富岳の富士通 A64FX CPU、インテル Sapphire Rapids CPU、NVIDIA H100 GPUがそれに該当する。個々のスレッドの遅延を低減する代わりに処理全体のスループットを向上させることに主眼を置いたアーキテクチャであり、そのために、容量を多少犠牲にしながらもHBMなどの広帯域メモリを採用する。スループット指向により遅延の削減機構に多くの資源を割かずに済むため、電力性能比はLSよりも高く、結果として高いピーク性能を有する。CCはメモリ帯域よりも演算そのものの性能を高めるため多数の演算を集積したアーキテクチャであり、その目的のために、特定の計算処理や対象とするアルゴリズムドメインに特化したハードウェア構成を有することが多い。例えば、Google TPUや、NVIDIA V100 GPUから搭載され始めたTensor コア、あるいはSystolic arrayやCoarse grained reconfigurable array (CGRA)といった、処理の空間的並列性や時間的並列性（パイプライン処理）を最大限に引き出すようなアレイ型のアクセラレータがCCに該当する。特化型のハードウェア構成により、高いピーク性能や電力性能比を持つことが可能である。

図 2.1.2.1に示す通り、メモリ帯域に性能が制約されるメモリバウンドから演算のみに性能が制約されるコンピュートバウンドまでの広範なアプリケーションに対応するには、LS CPUのみの構成では不十分であり、BC型ア

ーキテクチャを導入しながらも必要に応じて不足するコンピュートバウンドの演算性能を補うCC型アーキテクチャを組み合わせる構成が有望であると考えられる。ノードアーキテクチャの設計では、組み合わせる基本アーキテクチャの種類と、アーキテクチャごとの性能および電力のバランスが重要な指針となる。例えば、富岳A64FX CPUのような広帯域メモリを搭載したBC CPUにGPUやあるいはアレイ型のCCアクセラレータを組み合わせるといった構成が検討の対象となり得る。これらの背景を踏まえ、BC型やCC型アクセラレータ、それらを搭載したノードアーキテクチャやCPU接続の技術動向を以下にまとめる。

- アクセラレータ

スーパーコンピュータに対する要求性能と利用可能な電力容量の制限、昨今の脱炭素化へ向けた動向等からスーパーコンピュータの電力効率の向上は喫緊の課題であり、その解としてGPUをはじめとした演算加速装置 (アクセラレータ) の活用が現在の主流となりつつある。その証左として、図 2.1.2.2に示す2022年11月のGreen500の上位10システムは全てアクセラレータを搭載しており、それらシステムの殆どがNVIDIAもしくはAMD製のHPC向けGPUを採用したスーパーコンピュータであるのが現状である。

The GREEN 500

MOST ENERGY EFFICIENT ARCHITECTURES

TOP 500

Computer		Interconnect	Accelerator	Rmax/ Power
Henri, Lenovo ThinkSystem SR670 V2	Intel Xeon 32C 2.8GHz	Infiniband HDR	NVIDIA H100	***65.1**
Frontier TDS, HPE Cray EX235a	AMD EPYC 64C 2.0GHz	Slingshot-11	AMD Instinct MI250X	***62.7**
Adastra, HPE Cray EX235a	AMD EPYC 64C 2.0GHz	Slingshot-11	AMD Instinct MI250X	***58.0**
Setonix-GPU, HPE Cray EX235a	AMD EPYC 64C 2.0GHz	Slingshot-11	AMD Instinct MI250X	**57.0**
Dardel-GPU, HPE Cray EX235a	AMD EPYC 64C 2.0GHz	Slingshot-11	AMD Instinct MI250X	**56.5**
Frontier, HPE Cray EX235a	AMD EPYC 64C 2.0GHz	Slingshot-11	AMD Instinct MI250X	**52.2**
LUMI, HPE Cray EX235a	AMD EPYC 64C 2.0GHz	Slingshot-11	AMD Instinct MI250X	**51.4**
Atos THX.A.B, BullSequana XH2000	Xeon 32C 2.4GHz	NVIDIA HDR100	NVIDIA A100	***41.4**
MN-3, Preferred Network MN-Core Server	Xeon 24C 2.4GHz	RoCEv2/MN-Core DirectConnect	MN-Core	**40.9**
Champollion, Apollo 6500	AMD EPYC 64C 2.45GHz	Mellanox HDR	NVIDIA A100	**38.6**

[Gflops/Watt]

図 2.1.2.2　SC22 (Green500 BoF) 発表資料

1位を獲得したのは、ニューヨークのフラットアイアン研究所にLenovoによって導入された「アンリ（Henri）」であり、このシステムが搭載するH100 GPU (アーキテクチャ名：Hopper) はNVIDIAが現在 (2023年4月) 提供している最新型GPUである。本GPUの製造プロセスルールは4nmであるため、7nmプロセスを採用した前世代のNVIDIA A100と比較して演算コア数は22%増加、動作周波数は約1.3倍向上している。その恩恵によって、IEEE FP64およびFP32のピーク演算性能はそれぞれ30 TFLOPS、60 TFLOPS (SXM5)に達しており、これらはA100の約3.1倍の性能である。また、これらの性能をTDPに基づいた電力対性能に換算するとH100 (TDP：700W)はA100 (TDP：400W)の約1.8倍の電力効率を達成している。さらに、今日の深層学習やAIワークロードの高速化の需要の高まりから、NVIDIAはH100の2世代前のGPUであるV100からTensorコアと呼ばれる行列積和演算に特化した演算回路を新たに搭載している。V100ではFP16精度と、Turing Tensorコアで追加されたINT8、INT4、バイナリ1ビット精度のデータ型をサポートしていたが、A100からはそれらに加えてTF32、BF16、FP64 形式のデータ型がサポートされ、

H100のTensorコアによるFP16、TF32、FP64のピーク演算性能はそれぞれ1000、500、60 TFLOPSにまで達している。なお、これらの性能とTensorコアを使用しない場合の性能差はそれぞれ8.3、8.3、2倍となる。

2位から7位には2022年6月、 11月のTOP500で首位となっているスーパーコンピュータFrontier (米国オークリッジ国立研究所)と同じ構成のシステム群がランクインしている。システムに搭載されたアクセラレータはAMDの提供するHPC向けGPUであり、MI250Xは現時点 (2023年4月) において最上位モデルとなる製品である。製造プロセスルールは6nm、IEEE FP64およびFP32のピーク演算性能はどちらも47.9 TFLOPSに達しており、FP64の理論ピーク性能に限って言えばNVIDIA H100 GPUを上回っている。また、TDPは500W (最大560W) であるため、MI250Xの電力効率は理論的にはNVIDIA A100 GPUの約3.2倍であり、かつMI250Xを搭載したFrontierベースのシステムはGreen500ランキングの大半を占めていることから、その電力効率の良さを実際に示している。そして、AMD MI250X GPUにおいても深層学習やAIワークロードの高速化の需要に対応すべく、行列積専用命令をサポートするCDNA 2アーキテクチャを採用しており、これによるFP64行列積のピーク演算性能は95.7 TFLOPSにまで達する。

NVIDIAやAMD以外に、IntelもHPC用途のGPUを開発 (コードネーム：Ponte Vecchio) しており、米国アルゴンヌ国立研究所に導入予定のスーパーコンピュータAuroraにアクセラレータとして搭載される計画である。GPU以外のBCアクセラレータとしては、NEC SX-Aurora TSUBASAシリーズのようなベクトルプロセッサが存在し、図 2.1.2.1で触れたCC型アクセラレータとしては、Green500の9位にランクインしたMN-Coreをはじめとした深層学習やAIワークロードの加速を目的とするASICベースのアクセラレータや特定のニーズに合わせた演算加速を実現する回路を柔軟に実装出来るFPGAなどが挙げられる。CC型アクセラレータの中でも特に、Cerabras社はAIに最適化された演算コアとローカルメモリ全てを1枚の大型半導体チップに統合するという挑戦的なアーキテクチャを採用している。Cerabras社はこのアーキテクチャに基づいたチップをWSE - Wafer Scale Engineと呼んでおり、このWSEを搭載したシステムであるCS-1のFP32のピーク演算性能は3.3 PFLOPSにまで達している。CS-1は既に米国アルゴンヌ国立研究所をはじめとする様々な研究機関に導入されており、国内でも、2019年12月に東京エレクトロンデバイス株式会社が Cerebras CS-1の代理店契約を正式に締結し、受注を開始している。そして、WSEの第二世代であるWSE-2を搭載したCS-2も既に稼働しており、CS-1の最大2倍の性能を発揮することが見込まれている。

先述したように、スーパーコンピュータの開発において現時点における最大の問題は電力問題であり、いかにして電力効率を向上させていくかが重要である。そのための手段として最有力と目されているのがGPU等のアクセラレータの活用であり、Green500はその証左となっている。従って、今後においてもヘテロジニアスコンピューティングのためのハードウェアおよびソフトウェアの研究開発はますます重要になることが予想される。

昨年度に続く調査の結果、2023年11月のGreen500では、1~5位のシステムは2022年11月時点でのランキング順と同じであることに対して、6位以下のシステムは図 2.1.2.3に示すような変動が見られた。具体的には、6位にはNVIDIA H100 GPUを搭載したMareNostrum 5 ACC、9位と10位にはそれぞれAMD MI210 GPUを搭載したGoethe-NHRとNVIDIA H100 GPUを搭載したOlafが新しくランク入りし、LUMIのシステム規模の増強による性能向上からFrontierとの順位の入れ替えが起こった。図 2.1.2.3に示すとおり、全てのシステムがNVIDIAもしくはAMDのGPUを採用しており、高い電力効率を達成するための手段としてGPUが最有力と注目されていることが分かる。したがって、アクセラレータの技術動向の中でもGPUテクノロジのトレンドは常に押さえておかなければならず、それが本プロジェクトの成功の鍵となると考える。

Rank	Computer	Acccelerator	Perf. per watt [GFLOPS/W]
1	Henri Lenovo ThinkSystem SR670 V2, Intel Xeon Platinum 8362 32C 2.8GHz, Infiniband HDR	NVIDIA H100 80GB PCIe	65.40
2	Frontier TDS HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, Slingshot-11	AMD Instinct MI250X	62.68
3	Adastra HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, Slingshot-11	AMD Instinct MI250X	58.02
4	Setonix – GPU HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, Slingshot-11	AMD Instinct MI250X	56.98
5	Dardel GPU HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, Slingshot-11	AMD Instinct MI250X	56.49
6	MareNostrum 5 ACC EVIDEN BullSequana XH3000, Xeon Platinum 8460Y+ 40C 2.3GHz, , Infiniband NDR200	NVIDIA H100 64GB	53.98
7	LUMI HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, Slingshot-11	AMD Instinct MI250X	53.43
8	Frontier HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, Slingshot-11	AMD Instinct MI250X	52.59
9	Goethe-NHR Supermicro AS-4124GS-TNR, AMD EPYC 7452 32C 2.35GHz, Mellanox InfiniBand EDR	AMD Instinct MI210 64 GB	46.54
10	Olaf Lenovo ThinkSystem SR675 V3, AMD EPYC 9334 32C 2.7GHz, Infiniband NDR 400	NVIDIA H100	45.12

図 2.1.2.3　2023 年 11 月の Green500 ランキング
(https://www.top500.org/lists/green500/2023/11/)

- ノードアーキテクチャ・CPU 接続

2023年時点では、アクセラレータを搭載したノード構成においては、アクセラレータを制御するCPUはx86のISAをサポートしたメニーコアが広く用いられており、これがノードあたり1～2 個搭載されている。ホストメモリとして現在広く用いられているのはDDR規格のDRAMで構成されるデバイスであり、CPU (ソケット) あたりにDDR4メモリモジュールが複数チャネル接続される構成が一般的である。例えば、ソケットあたりに32 GiB DDR4-3200メモリモジュールが8チャネル接続される場合、メモリサイズは256GiBであり、メモリバンド幅は204.8 GB/sとなる。ただし2.1.2.3で述べられている様に、2021年末から第5世代の規格であるDDR5メモリの発売が開始されており、現在筑波大学計算科学研究センターで運用中のスーパーコンピュータPegasusでは、ノードあたり (CPUあたり) に16 GiB DDR5-4400メモリモジュールが8チャネル接続されている。従って、メモリサイズは128GiBであるが、メモリバンド幅は281.6 GB/sとなる。

CPUとアクセラレータ間のインターフェイスとして広く用いられているのはPCIeであり、NVIDIAが現在(2023年4月) 提供している最新型GPUであるH100 GPU (PCIe版)は5.0規格をサポートしている。また、PCIeはノード間接続のためのネットワークカード (例：InfiniBand HCA) やNVMeのインターフェイスとしても用いられるため、ノード内のPCIeデバイスはPLXと呼ばれるPCIeスイッチを介して接続される場合が多い。PCIe以外のインターフェイスとしては、NVIDIAが提供するNVLinkやAMDが提供するInfinity Fabric

があり、前者はスーパーコンピュータSummit (米国オークリッジ国立研究所)、後者はスーパーコンピュータFrontier (米国オークリッジ国立研究所) で採用されている。これらのテクノロジは同世代のPCIeインターフェイスと比較して数倍以上の帯域幅を有する以外にもキャッシュコヒーレンシをサポートすることからGPUプログラミングが簡素化されるという利点があり、アプリケーション開発者の注目を集めている。そして現在、CPUとGPU を1パッケージ化して両者を更に密結合させるアーキテクチャについてNVIDIA、AMD、Intelの各社は注力しており、NVIDIAはGrace-Hopper Superchip、AMDはMI300、IntelはFalcon Shoresのリリースを計画している。

昨年度に続く調査の結果、上記の技術動向に関して本年度は以下の更新が見られた。2023年11月13日に最先端共同HPC基盤施設（JCAHPC: Joint Center for Advanced High Performance Computing）は、NVIDIA GH200 Grace Hopper Superchipを搭載するスーパーコンピュータシステム OFP-II（仮称）の仕様を公開した。GH200はArm Neoverse V2のIPライセンスを利用した72コアのArm CPU(Grace)と、NVIDIAのNVIDIA H100 GPU(Hopper)を1モジュール上に統合したコンピューティングモジュールであり、CPU用に120GB (OFP-IIは120GB版を採用している) のLPDDR5Xメモリ、GPU用に96GBのHBM3メモリをそれぞれ搭載している。そして最大の特徴は、CPU・GPU間のインターコネクトにNVIDIA NVLink-C2Cを採用しており、450GB/sの帯域を確保しながらCPUとGPUのそれぞれのメモリにあるデータにそれぞれの計算デバイスが透過的にアクセスできる仕組みとなっている点である。OFP-IIの運用は2025年1月を予定しており、現在は導入作業を実施している。

2.1.2.5　ネットワーク

昨年度の報告書では、以下の調査結果が記されている。オフチップ相互接続ネットワークは、今後もHPC システムの主要なボトルネックとなる可能性が高い。データセンタやクラウドにおけるAIワークロードの急増により、より高速で緊密に結合したネットワークを必要とするのはHPCだけではなくなりつつある。一方で、従来型の性能向上では、今後数年間で帯域幅は良くて4～8倍程度、遅延は現状維持または悪化する可能性がある。

HPCシステムの大規模な相互接続ネットワークでは、性能やコスト、消費電力を決定する重要な要素がシリアライザ／デシリアライザ (SerDes) である。現在、InfiniBand NDR、Omni-Path (25G)、100G Ethernetなど、1リンクあたり100Gbpsまでの高速シリアル伝送が主流で、ケーブル1本あたり4リンクが一般的である。したがって、25GbaudのNRZ変調 (または伝送損失が若干悪化する50GbaudのPAM4変調) で、最大400Gまで利用可能である。5440ビットのブロックサイズを持つFEC(Forward Error Correction) は、PAM4のビットエラー率を低下させ、約100nsのオーバーヘッドで最大15のシングルビットエラーまたは最大150ビットのバーストエラーに対応できる。今後、25/50Gイーサネットは、より低遅延の2720ビットFECに移行する可能性がある。今後10年で、InfiniBandはXDR (4xリンクで800Gbps)、GDR (1.6Tbps) を目標とし、イーサネットも同様である。

電気信号の帯域幅は距離が長くなるにつれて狭くなるが、光信号の伝送 (ファイバー使用) は距離にほぼ影響されない (0.05dB/cm)。CPO (Co-packaged Optics) とSiPho (シリコンフォトニクス)は、光と電気 (OE) の変換をLSIに近づける技術であり、SiPhoによる現在のスイッチASICの容量は25.6Tb/sに達している。CPOは、さらなる容量増加、30%以上の消費電力削減、OE変換の高速化、パッケージ密度向上等をもたらす可能性があるが、パッケージング等に研究課題が残る。

上記のSerDes、トランシーバ、ワイヤ等の基盤に基づき構成されるHPCネットワーク技術の具体例を以下に述べる。富士通のTOFUは、カーネルバイパス、RMDA、MPIをサポートし、バリアとリダクションを実行する専用のネットワーク内処理ハードウェアを必要とする。現在、ノードあたりの帯域幅は40.8GB/sまで向上し、1ホップの遅延は700ns以下まで低下している。TOFUの強みは、電力効率と低遅延だが、トポロジの制限やCPUダイの

専用設計が必要な点が課題である。オープンな仕様に基づくInfiniBand (IB) は、何世代にもわたってHPCハードウェアとソフトウェアを提供している。現在、NDR (50GB/s、900ns以下のレイテンシ) は、カーネルバイパス、RMDA、MPIに加え、広範なネットワーク内処理と独自のアダプティブルーティングをサポートしている。IBの間接ネットワークは、専用のスイッチ (最大64ポート) を必要とするが、多くのトポロジ (FatTree、Dragonflyなど) を公式にサポートし、理論的には全てのトポロジをサポートできる。IBの帯域は3年ごとに約2倍のペースで向上しているが、コストの上昇、レイテンシの停滞、NIC/スイッチの消費電力が懸念材料である。Ethernetはカーネルバイパスと RMDAを備えていないが、価格、相互運用性、ソフトウェアサポートの点から、ローエンドのHPCシステムやデータセンタへの採用が多い。InfiniBandと比較すると、Ethernetは帯域幅では同等だが遅延はやや劣る。RDMA over Converged Ethernet (RoCE(v2)) は遅延の問題を緩和するが、採用例は今のところ多くない。EthernetのNICとスイッチによるエネルギー消費は、性能、機能、プログラマビリティに大きく依存する。HPEのSlingshotは、IBとEthernetの強みを組み合わせようとするものである。ソフトウェアはTCP/IPとRDMAを簡単に切り替えることができ、スイッチはそれに応じてパケットを処理する。また、QoS、ロードバランシング、アダプティブルーティング、輻輳管理などの機能が利用できる。現在、Slingshotは今のところ最大25GB/sのdragonflyしかサポートしておらず、加えて遅延の大きさも懸念される。Compute Express Link (CXL) とNVLinkは、同一ノード内のデバイス間の高速インターコネクトを目的として始まったが、今ではスケールアウト機能を持つ小規模なノード間接続をもターゲットにしている。Rockportはパッシブ光スイッチを提案しているが、ファイバーシャッフルのため大規模化に課題がある。EXTOLLはTOFUと似ているが、実際のシステムに使われた例は少ない。Bull eXascale Interconnect (BXI) はIBに似ているが、これも大規模な設備ではほぼ採用例がない。

ここまでネットワークシステム全体をカバーする既存の技術について述べたが、最適なシステムの構築にはSiPhoの進展やトポロジにも気を配る必要がある。現在主流となっているプラグイン光伝送の電力効率は約25pJ/bitだが、次世代HPCではSiPhoの利用により大幅な削減が期待される。すでにSiPhoを用いた光スイッチのプロトタイプでは、1pJ/bit程度を実現しており、今後はOBO (基板上モジュール) 、CPO (パッケージ内モジュール)、OE (チップ内変換) の順に技術が進展すると考えられる。必要な電力は、OBOで20pJ/bit程度、CPOで5pJ/bit程度と予想されている。一方、製造コストと光スイッチングという2つの課題がSiPhoと全光インターコネクトの普及を阻んでいると考えられる。光のバッファリングや光子のヘッダー処理、電気-光変換の低コスト化等の問題から、複雑なトラフィック制御、スイッチング等による遅延増加、リンクの活用不足など様々な欠点を持つ回線交換を選択する可能性もある。トポロジに関しては、メッシュ、トーラス、ハイパーキューブ、Clos、butterfly、dragonfly、slimfly、jellyfish等が、HPCやデータセンタをターゲットに研究されてきた。しかし、多くのHPCシステムで採用されるClosの派生型や、TOP500等のハイエンドに多いトーラスやdragonfly等、実際に採用されているトポロジは限られる。理論的には、スイッチに十分なポート数があれば全てのトポロジを実現できるが、システムスループットとコストの最適化が重要となる。

ネットワークに対する代表的なアプリケーションからの要求を考えると、3次元物理シミュレーションにおけるレイテンシ重視の小メッセージの近傍交換、AIワークロードにおける帯域幅重視の大容量削減、FFTにおける小/中サイズのオールツーオール通信等が挙げられるが、これらのすべてに適したトポロジはない。遅延に敏感なアプリは最小限のネットワークホップまたは低径ネットワークを必要とし、一方でall-to-all通信は完全な二分バンド幅を必要とし、帯域幅に敏感なアプリは通信パターンにもよるものがその中間である。現実的なアプリケーションはシステム全体の一部により実行されるためネットワークの縮小は可能だが、ジョブ間およびPFSの干渉について考慮する必要がある。

データセンタのネットワークは従来、信頼性、保守性、拡張性、柔軟性を重視し、HPCは低遅延と高帯域幅を重視してきた。現在では、AIやローエンドHPCのクラウド化が進み、これらのネットワークの機能が融合しつつあ

る。そこでは、セキュリティ、仮想化、光スイッチ、標準化設計、相互運用性の重要性が増している。プログラマブルSmartNIC (FPGAやDPUを搭載) は、両者のニーズを満たすことが可能だが、エネルギー効率の向上やコストの削減が必要になる。

将来のHPCシステムにおけるネットワークは、CPUとアクセラレータの緊密な結合、低遅延、様々なネットワーク機能 (カーネルバイパス、HWオフロード、ネットワーク/キャッシュ結合、ネットワーク内処理、暗号化など) 等の要素をリーズナブルなコストで実現することが求められる。SiPhoとSmartNICの組み合わせは、考えられる有効な1つの方法である。他にも、CXL、NVLink、またはロックポートのソリューションなどを使用して、高帯域幅のアグリゲータースイッチを備えたゼロ/低パワーの「ポッド」からなる階層型ソリューションを構築することや、低遅延で安価な電気パケットスイッチと帯域を確保する光回路スイッチを組み合わせたセミ・マルチレール・ネットワーク等も考えられる。いずれにせよ、大規模な投資と共同設計が必要となる。イーサネットと1.6Tbpsを待つだけでは、現在のシステムと比べて大きなメリットは得られないと予想される。

本年度は、UEC (Ultra Ethernet Consortium) に参加し、次世代のHPCおよびAIが要求する広帯域通信の実現を目的とするイーサネットのロードマップ調査を行った。また、長時間スケールの並列離散イベントシミュレーション(PDES) とサロゲートモデルによる学習とを組み合わせた、ハードウェア協調設計シミュレーションを高速化するプロジェクトにも参加した。これらのプロジェクトは開始後間もないこともあり、現時点では大きな成果は見られないが、今後もこれらのプロジェクトに加えて、次世代のHPCやAIのための大規模システムのネットワーク技術動向についての調査を継続する。

2.1.3 ワークロード分析

2.1.3.1 分析の概要

各アプリケーションサブグループにヒアリングを行った結果から、今年度のアーキテクチャ評価において優先すべきベンチマークコードとして以下の8本を選定した。

- GENESIS
- SALMON
- SCALE-LETKF
- EbE-method
- FrontFlow/blue
- LQCD-DWF-HMC
- Hugging Face GPT-2 XL
- Megatron-LM DeepSpeed

選定の基準や各ベンチマークの詳細については、4.11.1.2を参照されたい。

2.1.3.2 各ベンチマークの概要

選定した8本のベンチマークのそれぞれについて、特徴などの分析を行った。

2.1.3.2.1 GENESIS

- **概要：**GENESIS は生体分子の分子動力学計算ソフトウェアであり、ポスト「京」の重点課題アプリケーションの 1 つに選定されていたアプリケーションである。Intel CPU や A64FX 等の各アーキテクチャ向けに最適化されたコードが存在するが、本 FS のベンチマークとして提供されたコード（以下、FS 向け

GENESIS）は Intel CPU 向けのコードの主要な計算部分を抽出したものとなっている。コードは Fortran 90 で記述されており、主要な計算は単精度浮動小数点で行われる。

また、オリジナルのコードではOpenMPとMPIによるハイブリッド並列化やCUDA実装も行われているが、FS向けGENESISは1プロセス実行を想定しており、OpenMPによる並列化のみが行われている（CUDA版はない）。これは、GENESISの場合、原子数の増大によって増加するのは基本的にはプロセス数であり、各プロセスの計算の特徴はほとんど変わらない（弱スケーリングする）ためである。なお、FS向けGENESISの計算規模は約10万原子を16MPIで実行した時の1プロセス相当であり、10万原子は2030年に想定される計算規模の1/10である。

GENESISにおいてはPairlistとKernelの2つの計算カーネルが性能上のボトルネックとなる。ただし、Pairlistの実行は数ステップ（富岳開発時の想定は10ステップ）に1回なのに対し、kernelの実行は毎ステップ必要なことから、PairlistよりもKernelの方が相対的にボトルネックである。

- **Pairlist**：短距離・非結合性相互作用を計算する原子ペアの数え上げを行う。原子の座標を表す 3 次元配列（SoA）へのアクセスと原子間の距離計算（浮動小数点加算、乗算）が主な計算内容である。カーネル本体のコード行数は 240 行程度と短い。3 次元配列へのアクセス開始位置は間接参照だが、そこから連続するデータに対してアクセスが行われる。A64FX における B/F は 1.4-22.5、配列の間接参照があるため SIMD 化率は 7.3%（ピーク性能比 3.9%）と低めである。キャッシュブロッキング等の最適化は行われていないが、L1D キャッシュのヒット率は非常に高い（A64FX では 99%超）。ループの深さは最大 4（最内ループに条件分岐あり）であり、最外ループが OpenMP で並列化されている。A64FX では命令フェッチ待ち、L1D キャッシュのアクセス待ち、浮動小数点演算待ちが多い。
- **Kernel**： Pairlist で数え上げた結果を元に短距離・非結合性相互作用力を計算する。5 種類の 3 次元配列と 1 種類の 4 次元配列（SoA）を計算に使用し、浮動小数点加算と乗算が主に行われる。カーネル本体のコード行数は 190 行程度と短い。主要な配列へのアクセスはそのほとんどが連続である。A64FX における B/F は 0.36-5.8、SIMD 化率は 73.9%と高めであるが、ピーク性能比は 5.2%と低い。Pairlist と同様、キャッシュブロッキング等の最適化は行われておらず、A64FX では L1D キャッシュのミス率が 8.9%とやや高めである（ただし、ほとんどのアクセスは L2 までにヒットする）。ループの深さは最大 2（最内ループに条件分岐無し）であり、最外ループが OpenMP で並列化されている。A64FX では L1D キャッシュと L2 キャッシュのアクセス待ちが多い。

2.1.3.2.2 SALMON

- **概要：**SALMON は電子ダイナミクスの量子力学計算ソフトウェアであり、本 FS のベンチマークとして MPI+OpenMP で並列化されたフルアプリケーションコードが提供されている（OpenACC+CUDA 版もあるが、一部の環境でしか動作確認ができていない）。SALMON は様々な計算が可能であるが、本 FS では基底状態計算（GS）と時間発展計算（TE）の 2 種類の計算を評価対象としている。コードは基本的には Fortran 90（一部は C 言語）で記述されており、主要な計算は complex 型で行われる。
- **特徴**：GS と TE ともにステンシル計算部分が性能上のボトルネックである。A64FX においてステンシル計算の実行時間がアプリケーション全体の実行時間に占める割合は、GS の場合は 25%、TE の場合は 93%にも達する。ステンシル計算は 3 次元、24 点（X, Y, Z の各軸に対して-4, -3, -2, -1, 1, 2, 3, 4 の 8 点）のデータに対して行われる。ステンシル計算部分は逐次コードで記述されており、ステンシル計算の外側で OpenMP による並列化が行われている。2 種類の 5 次元配列を計算に使用しており、配列へのアクセスは直接参照かつ規則的である。A64FX における SIMD 化率は GS で 46%、

TE で 66%であり、L1D キャッシュ、L2 キャッシュともにヒット率は 97%を超える。また、A64FX では同期待ち（GS のみ）、浮動小数点演算/ロード待ち（GS と TE 共通）が多いのが特徴である。

2.1.3.2.3 SCALE-LETKF

SCALEは気候・気象科学および計算機科学の専門家らにより開発された次世代の気象気候科学におけるオープンソースの基盤ライブラリである。大規模なスーパーコンピュータから一般的な汎用サーバまで、様々な環境における利用を想定して開発された。一方、LETKF (Local Ensemble Transform Kalman Filter) は局所的にアンサンブル変換カルマンフィルタを用いたデータ同化手法であり、汎用的な Fortran モジュールと、Lorenz-96 や Speedy、WRF などいくつかのモデルへの実装が公開されている。SCALE-LETKFは、SCALEの領域を限定したモデルであるSCALE-RM (Regional Model) とLETKFによるデータ同化処理とを連携したものである。各プログラムは別々に実行され、シェルスクリプトによって連携して動作する構成となる予定である。

提供されたプログラムはFortran90で書かれており、先述の通りSCALE-RMとLETKFにコードが分かれているが、現在の実装では実際に実行されるのはLETKF部分のみである。メインの演算は倍精度浮動小数点で、特徴としてデータ同化処理の際に大量のデータを読み書きする必要がある。実行時の設定は、テスト用データサイズ約3.3GB、アンサンブル数11、12スレッド並列実行である。

実行時の特徴として、LETKF部分のみの実装であるためデータ入出力のコストが非常に大きい点と、ノード内の1つのコアに処理が集中している点、およびその他のコアは実行時間のほとんどが同期のための待ち時間となっている点が挙げられる。SCALE-RMと共に実行した場合には、現在の実装では有効に活用されていないこれらのコアによりSCALE部分の処理が実行されるものと考えられる。キャッシュミス率は非常に低くA64FX向けの最適化が施されていると考えられるが、データ入出力の割合が非常に大きく、演算性能よりもデータI/Oが性能に大きく影響している。現段階ではSCALE-RMを含む完全なアプリケーション実装が無い為、完全な実装を待って詳細な解析を行う必要がある。

2.1.3.2.4 EbE-method

Element-by-Element（EbE）法は、有限要素解析に必要となる行列演算を分解し、より小さいサイズの要素係数行列による演算を用いて、計算量を削減し並列化に適した形式に変換する手法である。EbE法のメリットとして、元の行列を直接用いる場合と比較して必要となるメモリアクセスが大幅に削減可能である点と、複数回の小さな要素係数行列の演算に分解されることから並列処理による高速化に適している点である。一方で、並列処理による高速化は、用いる計算機のアーキテクチャに応じた最適化が必要となる。例えば、要素係数行列データをキャッシュに格納できるかどうかで、処理速度は大きく変化する。

本プログラムでは、EbE法による2次四面体要素の行列-ベクトル乗算カーネルであるシングルノード版および有限要素解析アプリケーション全体を実装したマルチノード版が提供されている。本実装はFortran90で記述されており、A64FX向けの最適化が行われている。最適化の内容としては、アプリケーションのメインのループをデータの再帰が必要ない前半部分と必要な後半部分に分割して、前半の浮動小数点演算をSIMD化して高速化を実現している。また、ループブロッキングを用いることで、上記のSIMD演算をA64FXのL1キャッシュ上で実現可能なサイズに調整し、キャッシュミスによる性能低下を抑えている。

シングルノード実装はOpenMPによりスレッド並列化されており、12コアで並列処理した場合、SIMD化率は約40%で、L1キャッシュミス率は1%以下となっている。この結果から、L1キャッシュサイズに収まるようにループブロッキングが設定されており、A64FX向けに適切な最適化が行われていることが分かる。マルチノード実装はOpenMP化に加えてMPI化され、マルチノード実行が可能となっている。4ノード×12コアの場合、

SIMD化率は約55%に向上し、L1キャッシュミス率も2%程度に抑えられている。一方で、ノード間の同期のためのオーバーヘッド(実行時間の約25%)による効率の低下がみられた。しかしながら、ノード間の通信量は少ない(全体データ量の1-2%)ため、並列化により十分な高速化の効果が得られると考えられる。実際に、全く並列化されていないシングルスレッド実行による性能に対して、マルチノード(計48スレッド)での実行は30.7倍の高速化を実現している。

2.1.3.2.5 FrontFlow/blue

概要：FrontFlow/blue (FFB) は非定常非圧縮性 Navier-Stokes 方程式を時間、空間ともに 2 次精度を有する有限要素法により離散化した汎用流体解析コードであり、空力騒音解析やターボ機械内部流れ解析等の様々な産業分野で活用されている。領域分割による並列計算がサポートされており、圧力勾配ベクトルを計算して圧力方程式を解くサブルーチンが主要な計算カーネルとなる。

プログラムの特徴：FFBのソースコードの総行数は50,772であり、FORTRAN77で実装されている。先述したように、このアプリケーションの主要な計算カーネルは圧力方程式を解くサブルーチンであり、そのコールグラフを図 2.1.3.1に示す。図中のpres3eが圧力方程式を解くサブルーチンで、赤枠で囲んだサブルーチンに圧力勾配を計算するループが含まれている。

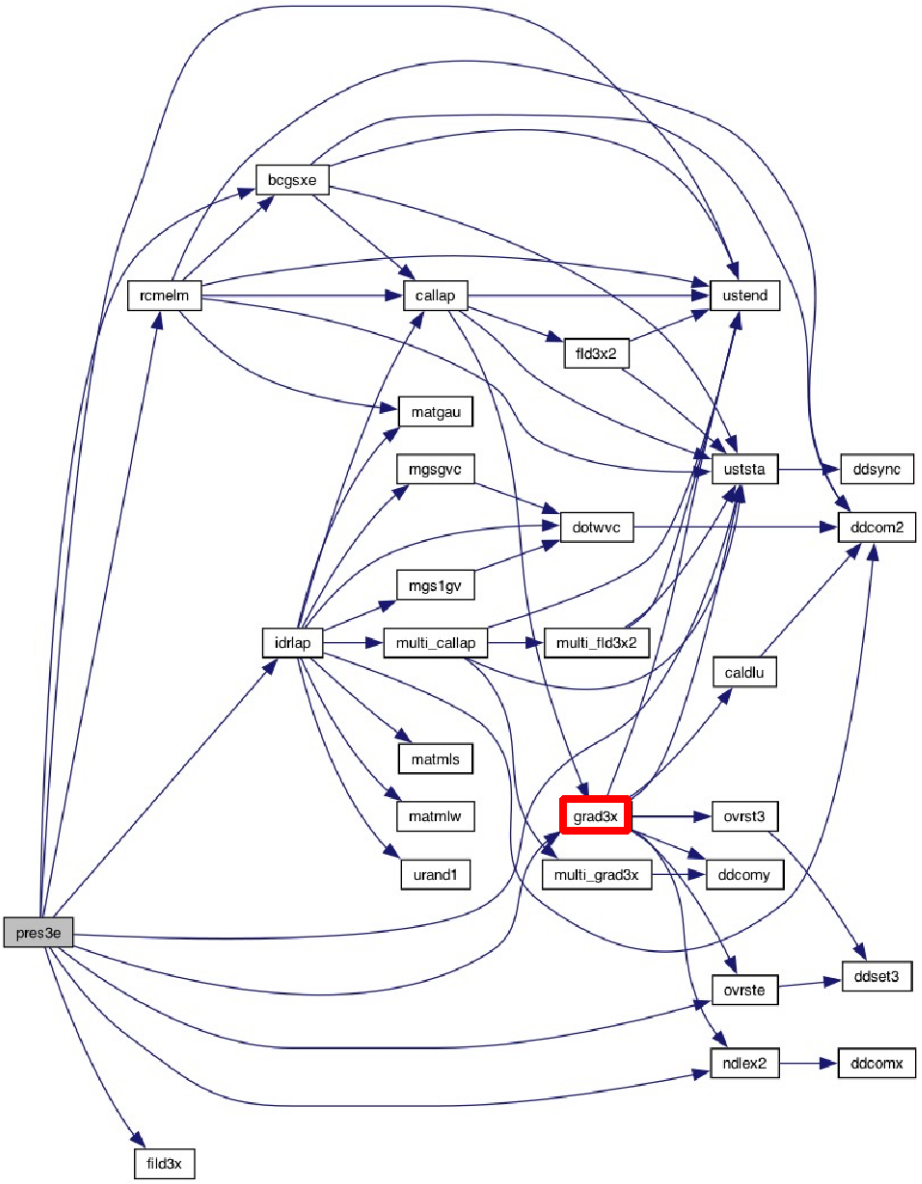


図 2.1.3.1 コールグラフ

図 2.1.3.2がそのループであり、四面体の圧力勾配ベクトルを計算している。この三重ループの外側・中間・内側のループはそれぞれ、隣接関係の全くない要素グループ (カラーと呼ばれる) 、特定の色のグループ内の要素セットを、セット内の要素の数を担当し、並列化はカラーリング (外側) をすることによって実現されている。そして、B/Fはメモリアクセス量が 16 (NODE, R) + 16 (SN, R) + 4 (VALELM, R) + 16 (VALNOD, W) = 52バイト (データ型は単精度浮動小数点)で、合計フロップ数 = 4（乗算） + 4（加算） = 8フロップであるため、6.5 (= 52 / 8)となる。したがって、並列化可能であるが、メモリバンド幅律速なアプリケーションであるため、いかにキャッシュメモリにデータを乗せるかが肝要となる。そして、CPU実行時の並列化はMPI+コンパイラによるスレッド並列、GPU実行時の並列化はMPI+OpenACC (オプション−Dgpudirect で、GPUDirect を有効化可能であり、CPU-GPU間のデータ管理にはUnified メモリを使用) を採用している。

```
C     * TET *
DO 1111 ICOLOR=1,NCOLOR(1)
!ocl norecurrence(VALNOD)
DO 1110 ICPART=1,NCPART(ICOLOR,1)
    IES=LLOOP(ICPART   ,ICOLOR,1)
    IEE=LLOOP(ICPART+1,ICOLOR,1)-1
!ocl nosimd
!ocl noswp
    DO 1100 IE=IES,IEE
        IF (LEFIX(IE).EQ.1) GOTO 1100
        IP1 = NODE(1,IE)
        IP2 = NODE(2,IE)
        IP3 = NODE(3,IE)
        IP4 = NODE(4,IE)
        VALNOD(IP1) = VALNOD(IP1) + SN(1,IE)*VALELM(IE)
        VALNOD(IP2) = VALNOD(IP2) + SN(2,IE)*VALELM(IE)
        VALNOD(IP3) = VALNOD(IP3) + SN(3,IE)*VALELM(IE)
        VALNOD(IP4) = VALNOD(IP4) + SN(4,IE)*VALELM(IE)
1100    CONTINUE
1110 CONTINUE
1111 CONTINUE
```

図 2.1.3.2 サブルーチンに圧力勾配を計算するループ

性能評価：A64FX(富岳)の 1 CMG = 12コア (1スレッド / コア) での評価結果を図 2.1.3.3に示す。得られた性能は151.74GFLOPSであり、理論ピーク性能の20.46%の実行効率であった。これは、先述したようにこのワークロードがメモリバンド幅律速であることに起因しており、計算コアではなくメモリがBusyであることは図 2.1.3.3からも明らかである。ただし、得られたメモリバンド幅は131.26 GB/sであり、1CMG での STREAM Triadが約 205 GB/s なので、キャッシュミス率を考慮すると、この評価で得られているメモリバンド幅は妥当だと考えられる。

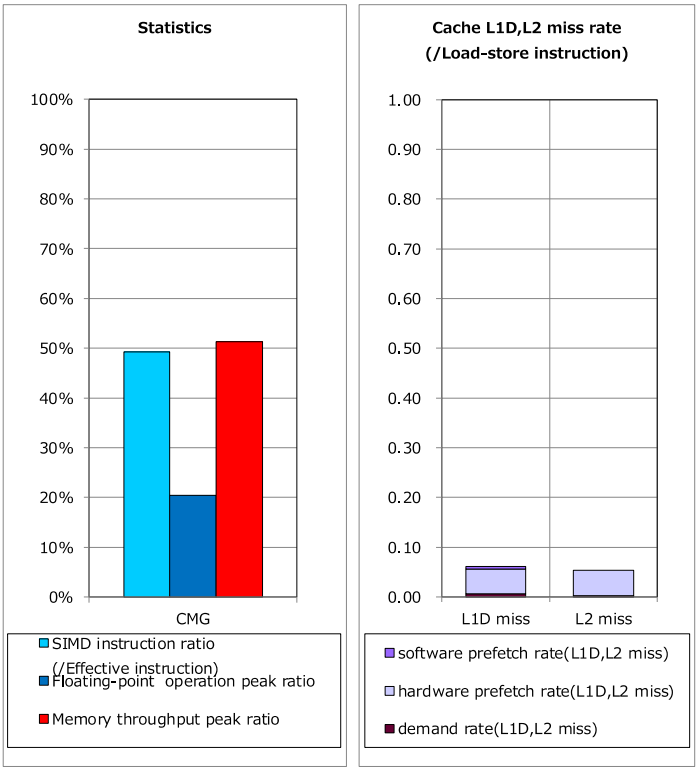


図 2.1.3.3 評価結果

2.1.3.2.6 LQCD-DWF-HMC

LQCD-DWF-HMCは富岳向けに最適化された格子QCDアプリケーションのカーネルライブラリ（QWS）をベースに、汎用性を高めたアプリケーションである。具体的には、QWSではSIMDを最大限にいかすためにステンシル計算のX方向（単精度）16個をベクトルレジスタに配置するデータレイアウトとしていたため、問題サイズxに32の倍数という制限があった。LQCD-DWF-HMCでは、XとYで4x4を配置するデータレイアウトにすることにより、問題サイズxが8の倍数、yが4の倍数と制約を緩和できている。性能は若干低下するが、問題サイズに対する強い制約はとれて使い勝手はよくなっている。それ以外はほぼ昨年度評価したQWSと同じである。

カーネル部と、Proxyアプリが含まれているが、Proxyアプリの入力データは実際的な値ではないため、各部の実行割合は実際とは異なることに注意が必要である。実際的なパラメータではカーネル部の割合が90%と見積もられる。以下では、カーネル部について検討を行う。GPU版も提供されているが、最適化はされておらず性能は高くない。

プログラム構造としては、Bicgstabによるソルバが支配的である。並列化は、MPI+OpenMPのハイブリッドで行われている。参照実行を行ったA64FXでは、4ppn (1rank = 1CMG, 12threads) が想定されている。512bit wide SIMDを意識したデータレイアウトがされており、L2キャッシュサイズを意識したキャッシュブロッキングや、L1ソフトウェアプリフェッチによる最適化などが行われている。通信もuTofuを利用し、通信と演算のオーバーラップを行うなど最適化されている。ビルド時のフラグでこれらのA64FX向け最適化はオフにすることも可能である。

問題サイズ32x8x8x12、グリッドサイズ1x1x2x2で、プロファイラを用いたA64FX1ノードでの評価を図2.1.3.4に示す。SIMD化率は60%と高い。演算効率は12%であるが、単精度が主なので、演算器効率としては6%である。実行サイクル割合としては、命令コミットが22%に対して、バリア同期が25%、L2 waitが13%、L1 waitが10%、Memory waitが7%、演算waitが16%であった。メモリ使用量は1.7GB、メモリbusy rateが25%であるが、キャッシュミス率はL1 11%、L2 3%であり、キャッシュブロッキングが有効に機能していると思われる。キャッシュミスのうち、デマンドミス率はL1、L2ともに 22%、ソフトウェアプリフェッチミス率はL1 37%、L2 20%であり、プリフェッチが有効に働いている。BF比は、命令数からは2.9 byte/floatと見積もられるが、メモリアクセス数からは、0.75 byte/float とキャッシュにより軽減していることがわかる。

言語はC++であり、ライブラリのソースコード(.cpp)は39,101行、ヘッダファイル(.h)は58,925行（ただし、ベンチマークでは使われていないルーチンを含む）で、ベンチマーク本体は合わせて1,557行である。

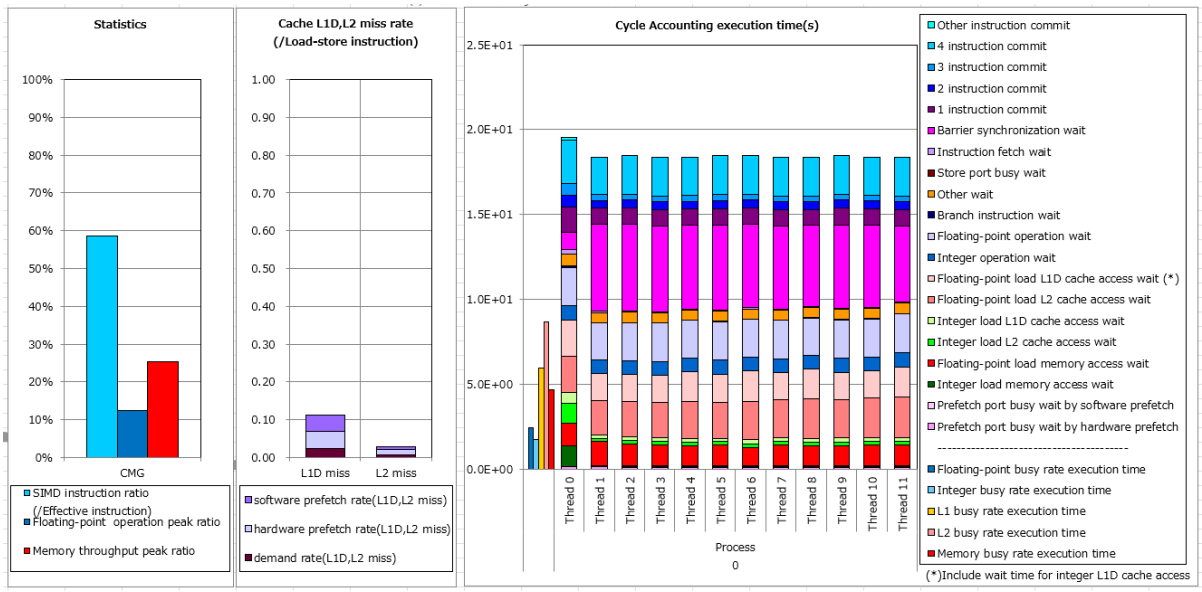


図 2.1.3.4 プロファイラを用いた A64FX1 ノードでの評価

2.1.3.2.7 Hugging Face GPT-2 XL

- **概要**：単一ノード版で提供されている。このベンチマークは外部依存関係を最小限に抑えるように設計されており、モデルは純粋な PyTorch で実装されている。提供されているバージョンでは、変更内容について厳しい条件が設けられている。モデルは、backward ステップなどのみ最適化できる。入力文字列から、次の単語を予測する Causal Language Model (CLM)の学習のスループットに関して、1 秒あたりのトークンで評価する。モデルは、LLaMa アーキテクチャに従って実装されている（rotary embeddings を使用）。GPT-2 XL (1.5B) と同様に、14 億 6,000 万個のパラメータを含むようにパラメータ化モデル定義は、すべての外部依存関係を取り除いた、Hugging Face トランスフォーマーのライブラリコードに基づいている。
- **詳細：**モデルは、 12 個の self-attention 層があり、それぞれ 32 個の attention ヘッド、埋め込みサイズ 3200、サイズ 6400 の混合層 がある。モデルは合成系列で学習され、それぞれ 256 トークン長がある。推論には、KV キャッシュ技術を使用し、運用環境での推論の通常の実行方法を模倣している。推論のバッチサイズは 1 に固定（デフォルト実行時）されている。
- **言語環境:** Python interpreter version 3.8 以上。標準 Python ディストリビューションに含まれるモジュール以外で必要な Python モジュールは「torch」のみとなっている。
- **実行条件：**実行時に、精度とバッチサイズを指定する。精度は、「FP32」、「TF32」、「FP16」、「BF16」が指定可能である。ベンチマーク報告には、サポートされているすべての精度の結果の明記が必要である。複数の数値形式を組み合わせて使用する場合には、たとえば、値は FP32 に保存され、累積演算はハードウェアの FP16 で行われる場合には、実行全体は最小ビット数の精度で行われたとみなされる。バッチサイズは、チューニングパラメタとする。
- **ベンチマークレギュレーション**：モデルは、以下の条件で、任意の方法でインスタンス化後にポスト処理ができる。本来の機能を維持しなくてはならない。同じ学習ループ内（例えば、Python インタープリタから呼ばれた「opimizer.step」)のモデル学習を許すこと。同じスクリプトを用いた実行となること。PyTorch フレームワークへの任意のセットアップステップの追加を許す。例えば「torch.backends.my_backend.enabled = True」など。新しい数値フォーマットの追加を許す。すべての変更は原則として「set_environment」および「set_model_and_data」メソッド内に

含める必要がある。コードの他の部分を変更する必要がある場合 (例: PyTorch API の変更など) は、ベンチマーク作成者へ連絡が必要。

- **コード所見**：図 2.1.3.5 に示すメインループに対する API 呼び出しの時間を高速化する必要がある。

```
• batch/__main__.py
  • def train(self):

    time_start = timer()
    for i in range(cnt_batches):
      self.net.zero_grad()
      batch = {“input_ids”:
        self.data, "labels": self.data}
      res = self.net(**batch)
      loss = res.loss
      loss.backward()
      optimizer.step()
    time_end = timer()

• net =
  LlamaForCausalLM(config)
```

図 2.1.3.5 メインループ

なお、batch/models/llama/modeling_llama.pyのclass LlamaForCausalLM (LlamaPreTrainedModel) は、PyTorch の CrossEntropyLoss を利用してlossの計算している。Optimizerとして、Pytorchのtorch.optim.AdamW (optimizer_grouped_parameters, lr=0.00001, eps=1e-06, weight_decay=0.01, betas=(0.9, 0.999)) を single optimization step として利用している。

- **性能実例**：名古屋大学「不老」TypeII サブシステムでの実行例を表 2.1.3.1 に載せる。なお環境は、以下の通りである。
  - singularity 3.7
  - PyTorch 2.1
  - 1CPU (Intel Xeon Gold 6230, 20 コア, 2.10 - 3.90 GHz × 2 ソケット)
  - 実行対象 run_prod_sample.sh

表 2.1.3.1より、バッチサイズの性能チューニングによる2.5倍ほど速度向上することが確認できる。

表 2.1.3.1 実行結果(FP32)

バッチサイズ	Tokens per second	倍率
1 (デフォルト)	107.5	1.00
4	216.0	2.00
8	269.7	2.50
16	268.4	2.49
32	ー(時間超過)	ー

2.1.3.2.8　Megatron-LM DeepSpeed

大規模言語モデルをMPIで分散処理するベンチマークである。元々は、GPUがメインのターゲットのようなコードの構成となっているが、CPUでも動くように移植され、富岳の環境でも動かすことができるように整備されている。プログラミング環境としては、PyTorchが使われており、C++やFortranが主流であるこれまでのHPCアプリケーションの構造とは大きく異なる。具体的な実測性能等は3.2.6節に記載している。

本ベンチマークは、Megatron-LMのコードベースに基づき構成され、テンソルやパイプライン並列もサポートされている。LLMモデルとしては、産業界でスタンダードでありプロダクションレベルで最も大きな単一モデルの実装ともいわれている GPT3-175B が用いられていて、1750億のパラメータの学習を行う。富岳で実行可能となる最小のノード数は富岳のネットワークトポロジも踏まえ384ノードに設定され、富岳の全系での実行にもスケールすることを目指して実装が進められている。

性能分析に関しては、C++やFortranが主流であるこれまでのHPCアプリケーションの環境から大きく異なるため、プロファイリングを行う設定や記述に関して独特のノウハウが必要であり、残念ながら今回のHugging Face GPT-2 XLの実行における性能プロファイリングを実施できるところに至らなかった。なお、現状では、PyTorchに実装されている標準のプロファイラ、富士通の提供するfipp基本プロファイラ、fapp詳細プロファイラがPythonの環境で利用可能であるとされている。一部のその分野のスペシャリストだけがアクセスできる環境という印象もあり、今後、利用者に向けた幅広い普及のためには使い勝手の向上などの課題があることが確認された。

2.1.4　アーキテクチャ検討の指針

2.1.4.1　アーキテクチャ検討の概要

昨年度においては、図 2.1.4.1に示す暫定版の性能・機能要件と、「計算性能（FLOP）ではなくデータ移動（Bytes）を指向し実効効率を高める”FLOPS to Bytes”コンセプト」をアーキテクチャの基本指針として、次世代計算基盤のシステム全体やその構成要素に関する技術的調査研究を行った。その結果、ベンダサブグループの提案する複数のアーキテクチャ候補と、その技術的可能性や到達可能な性能範囲などの情報が得られた。特に、CPUのみでは目標を達成する性能や電力性能比を得ることは困難であり、ベクトル演算性能やマトリックス演算性能に優れ電力性能比の高いアクセラレータと広帯域のメモリを有するアーキテクチャが有力な候補となることが確認された。

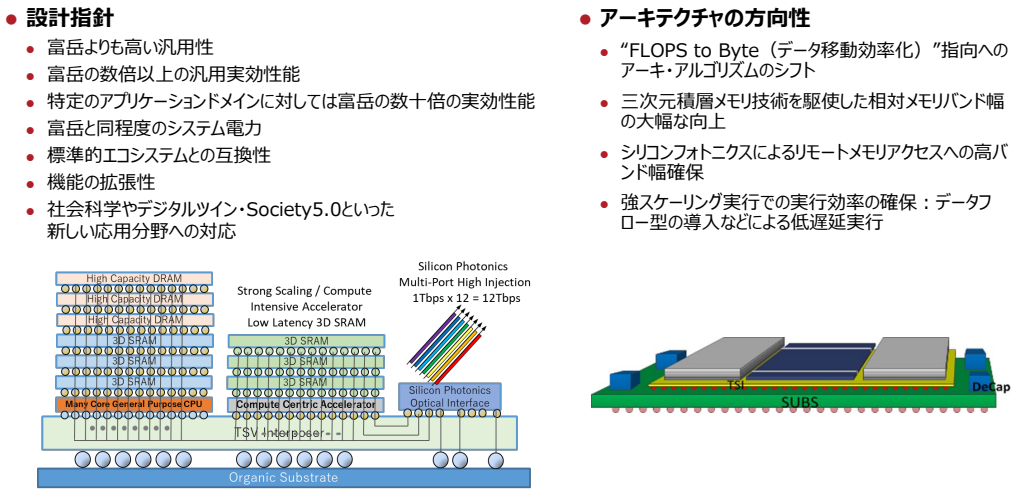


図 2.1.4.1　次世代計算基盤の設計指針とアーキテクチャの方向性

今年度では、それらに基づき、各ベンダの提案アーキテクチャ候補について、以下を実施した。

- アーキテクチャ候補の詳細化と、キーテクノロジの整理
  （マイクロアーキテクチャ、ノードアーキテクチャ、システムアーキテクチャ候補とその設計空間等の詳細化）
- アーキテクチャ候補に対する性能やコストの見積り
- アーキテクチャ候補に対するベンチマークの性能推定
- アーキテクチャ評価および比較による、有望な候補の絞り込み

2.1.4.2　評価項目案と各サブグループの検討内容

各ベンダサブグループから得られた複数のアーキテクチャ候補を比較するために、以下の評価項目を設定した。

- ノードアーキテクチャ
  - マイクロアーキテクチャやノード構成
  - 倍精度理論演算性能 [DP TF]
  - その他精度の理論演算性能
  - メモリ帯域 [TB/s]
  - Byte per FLOP
  - TDP [W]
  - 電力あたり性能 [GF/W]
  - アクセラレータと CPU の性能内訳
  - コスト
- システムアーキテクチャ
  - 電力制約下でのシステム規模
  - 電力制約下でのシステム理論性能
  - 理研の選定したベンチマークによる性能推定
- その他
  - 開発ロードマップ
  - リスクの高い技術要素

ベンダサブグループであるA2～5では、以上の項目に関するアーキテクチャ調査研究を実施した。また、それ以外のサブグループであるA1、6、7では、関連する要素技術や技術動向について調査研究を実施した。以下2.2節以降において、その詳細を述べる。

2.2 アーキテクチャ調査研究サブグループ 1（理研）

2.2.1 調査研究の概要

強スケーリング性能を有する、準汎用～専用加速装置アーキテクチャの検討を行っている。理化学研究所 生命機能科学研究センター 計算分子設計研究チームにおいて蓄積がある分子動力学シミュレーション専用計算機を題材に、まず専用加速装置としての開発を実施し、それをベースとしてその汎用化・準汎用化された演算加速装置の可能性を検討する。特に、分子動力学シミュレーション専用計算機の演算能力の中核であった非結合力計算の加速に加え、同期機構・メモリ内演算などの周辺回路における加速を重点的に検討すると同時に、これまで限定的にしか行っていなかった結合力計算の加速の検討を行っている。

今年度は、性能評価に向けた具体的な回路設計を中心に行うと同時に、LSI実装時の面積・電力評価、FPGAへの実装を行った。設計にあたっては、Chisel言語を使用し、極力パラメータ化して設計を行った。分子動力学シミュレーション専用計算システムで必要になる要素は以下である。

- 非結合力計算加速パイプライン
- 結合力計算・粗視化分子動力学計算向け汎用コア
- メモリ
- イベント制御部
- オンチップネットワーク
- 長距離力計算加速部
- オフチップネットワーク

具体的には、非結合力計算加速パイプライン・結合力計算・粗視化分子動力学計算向け汎用コア・メモリ・イベント制御部・長距離力計算加速部を一単位とする〝ComputeUnit (CU)〞を複数個オンチップネットワークで接続し、〝ComputeNode (CN, chipに相当)〞を構成する。ComputeNodeはオフチップネットワークを持ち、これにより相互接続されてシステムが構築される。

ほぼ必要な要素の設計を終え、現在デバッグを継続しながら評価を行っている。来年度は、引き続き設計を進め性能評価を中心に計算を行うと同時に、長距離力計算加速部を完成させる。これに基づき、汎用計算機に付加する専用アクセラレータの可能性、準汎用化の可能性の検討を進める。

2.2.2 アーキテクチャの検討状況

2.2.2.1 アーキテクチャ検討の方針

強スケーリング性能を有する、準汎用～専用加速装置アーキテクチャの検討を行う。我々の研究室での蓄積がある分子動力学シミュレーション専用計算機を題材に、まず専用加速装置としての開発を実施し、それをベースとしてその汎用化・準汎用化の可能性を検討する。

2.2.2.2 検討中のアーキテクチャ・テクノロジ・システムスタックなど

汎用計算部に付加する演算加速装置を検討している。今回の評価では、演算部のみでなく、同期機構・メモリ内演算などの周辺回路における加速を重点的に検討する。また、専用計算機の演算能力の中核であった非結合力計算の加速に加え、これまで限定的にしか行っていなかった結合力計算の加速の検討を行った。

2.2.3 調査結果

2.2.3.1 全体設計方針

今年度は、性能評価に向けた具体的な回路設計を中心に行うと同時に、LSI実装時の面積・電力評価、FPGAへの実装を行った。設計にあたっては、Chisel言語を使用し、極力パラメータ化して設計を行った。これは、

最適化のために加え、汎用化にあたっての設計流用を行いやすくするためである。

2.2.3.2　非結合力計算加速パイプラインの調査結果

非結合力計算加速パイプラインの実装は前年度の報告の通りMDGRAPE-4Aの設計を踏襲しChisel言語で開発を続けてきた。Chisel言語を用いることによりRTLのパラメータ化を施すことができた。これにより設計の再利用性が高まり、今回は特にパイプライン本数を調整しFPGAに載せることができた。パラメータ化した主な項目はパイプライン本数、キャッシュのウェイ数、計算した力を保持しておくRAMの深さである。パイプラインの内部モジュールもパラメータ化しており、今後は原子ペアを生成するモジュール等を、粗視化MD用の回路を実装する際に再利用できる見通しである。

またMDGRAPE-4Aの設計から以下の点を改善した。

(1) パイプライン本数をパラメータ化する際にパイプラインの稼働率の改善も行った。MDGRAPE-4Aの設計では計算する原子のIDの偶奇でデータを流す先のパイプラインを決定しており、非結合力のカットオフ長などの要因で計算するべき原子が偏ると、他のパイプラインが計算中であってもデータが流れずにアイドル状態に陥るパイプラインが存在するケースがあった。この偶奇で計算するパイプラインを決定するアルゴリズムは、パイプライン本数が4の倍数の時には効率が最大になるが、他のケースでは効率が落ちることもあったので、パイプライン本数をパラメータ化するのと合わせてアルゴリズムを見直して原子IDの偶奇に依らず全てのパイプラインに均等にデータを分配するように改良した。

(2) MDGRAPE-4Aでは原子ペアを生成する際にストールが発生していたが、アルゴリズムに修正を加えストールしないようにした。

非結合力パイプラインの準汎用化について、富士通研究所との共同研究により演算器アレイへの実装が可能であることを示した。

2.2.3.3　長距離計算加速部の調査結果

分子動力学計算における長距離計算は、クーロン力をカットオフ可能な二体間相互作用である近距離部分と、ゆるやかに変化する長距離部分に分け、長距離部分を空間の3次元ポアソン方程式により近似的に解くことで演算量を減らすエワルド法が一般的で、水分子など電荷の大きい原子を含む対象を陽に扱う場合には必須となっている。

PMEなどの格子を使う手法が主流で、1.粒子から格子点への電荷の補間、2.格子点電荷から格子点電位の計算（ポアソンソルバ）、3.格子点電位から粒子への力の補間、の3段階からなる。1.と3.の、粒子と格子点間の補間計算は演算と累積の繰り返しで、専用回路による加速が有効であるため、MDGRAPE-4Aの設計を踏襲してChiselによる書き直しを行った。

一方2)のポアソンソルバは、演算量よりも通信量とレイテンシに依存する計算であり、対象となる計算システムのサイズ、オフチップネットワークのトポロジによって最適な手法が変わってくる。まずは一般的な3次元FFTを使う実装で小規模の系を扱うことにより上記加速部の評価を始めているが、大規模な系では、演算負荷の大きい部分を加速したことにより、ポアソンソルバが最も時間がかかるようになると予想されるため、MDGRAPE-4Aで採用した畳み込み演算の加速による方法をより柔軟に実行できるよう、オフチップネットワークも含めた検討を進めている。

2.2.3.4　ComputeUnit(以下 CU)単体の調査結果

前年度開発を行った各要素、イベント駆動回路×1・専用パイプライン×4・汎用コア×1(4 threads)・メモリで構成されるCUを構築しシミュレーションにより動作検証した。最小の計算テストとして、水分子（3 atom, 2

bond, 1 angle, 3 nonbonded pair）のMD計算を回路シミュレーションで実施した。

また、構築した回路をFPGAで実装し、回路シミュレーションと同じ計算結果が得られることを確認した。FPGA実装結果に基づき回路の改良を行い、実装を効率化・タイミングの改善を図った。

2.2.3.5 結合力計算向け汎用コアのコンパイラの調査結果

前年度に、結合力や時間積分などの計算のための汎用コア（General Purpose core, GP-core）の開発を行った。これは通常の整数・浮動小数点の演算器のみならず、主に物理シミュレーションや機械学習などで用いられる数学関数の演算器も持ち、またhyperthreadingによって複雑なOut-of-Orderの機構なしにFPUや数学関数ユニットのパイプラインを全段稼働できるようになっている。

しかしながら、数学関数のレイテンシはFPUのそれよりも長く、データ依存が発生すると実行できない場合がある。この場合、アセンブラがエラー報告を行うものの、レイテンシを隠すための並べ替えは人力で行っていた。さらに、GP-coreが持つレジスタ数は32個と豊富ではあるものの、二面角計算などの一部の処理では足りなくなるため、人力で寿命を判定してアセンブラを書く必要があった。これを自動化するため、簡単なコンパイラを実装し、レジスタの割り付けとレイテンシの隠蔽を自動で行うようにした。

GP-coreは実装を軽量に保ち面積を小さくすることを第一とし、また制御機構の多くをイベントスケジューラに移譲しているため、関数の呼び出しでの文脈の退避や割り込み、リンクなどの機能を必要としない。また非常に多くの特殊な命令があり、どれも命令と一対一対応するような関数としてしか使用されない。よってコンパイラの実装にあたっては、それらの機能を実装する巨大な既存ライブラリを使用するのではなく、ごく小さなコンパイラを新規に実装した。その際、複雑になりやすく発見の難しいバグが入り込みやすいパーサーの実装を避けつつ、既知の言語を利用することで利用を簡単に、また最適化を容易にするため、C++言語上にDSLを実装し、C++で書かれた式と制御構造の情報を取得し、得られたASTを簡単な内部表現（IR）を経由してGP-coreのアセンブリへ変換するようにした。これにより、使用するレジスタ数を最小化し、命令数が最小となるような並べ替えを自動で行えるようになった。C++言語上にDSLを構築する機能は、GP-coreでの仕様変更に影響を受けないよう、また他の用途にも利用できるよう汎用のライブラリとして分離した。

また、このコンパイラでは、ジャンプ先にラベルを使用できることやレジスタ数に制限がないこと以外はほぼGP-coreの命令と違いのないIRを使用して、最適化パスの全てで途中のIR命令列を出力し、また最適化パスのそれぞれを独立したプログラムに分離することで、全ての段階で人間が介入できるようにした。これによって、自動的な並べ替えがうまく働かないような場合に、原因を特定し、あるいは回避することが容易になり、また新規な最適化を追加することも容易になった。

2.2.3.6 Network-on-Chip の調査結果

複数の計算ユニット間で計算に必要なデータをやり取りするため、Network-on-chip（NoC）の開発を行った。NoCのルータは以下のような要素を含む（図 2.2.3.1）。

- フリットの混雑によるスループットの低下を防ぐための、多重化された入力ポート
- 送信先の入力ポートをアロケートするモジュール
- 出力先の状態を使って出力の順序を制御する出力ポートとスイッチ制御モジュール
- 次のルータでの送信先ポートを計算するルーティングモジュール

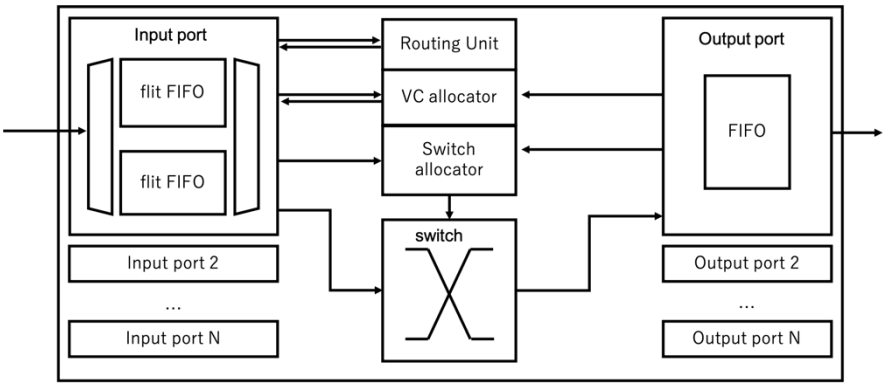


図 2.2.3.1　NoC ルータに含まれるモジュールとそれぞれの接続。

ルータの面積とレイテンシを小さくするため、接続は2Dメッシュトポロジとし、ルーティングアルゴリズムは他のルータの状態に依存しないstatic XY routingとした。さらにルーティングの際、他のルータとの依存がないことを利用して先んじて送信先でのルーティング結果を計算しておくことで、送信先のルータでのスイッチングにかかるレイテンシを削減した。これにより、ルータのレイテンシは2 cycleとなり、また計算ユニットと比較してFPGA上のリソース消費は1/500程度に抑えられた。

送信されるパケットはフリットに分割され、フリットに先頭と末尾のシグナルを付加することで、送信するパケットの長さに制限がかからないようにした。これにより、利用する計算ユニットの側でパケットのサイズを自由に変更できるようになる。さらに、計算ユニットとのインターフェイスでフリットのサイズをより細かく分割することで接続にかかる面積を削減する機能も実装した。この場合でもスループットは変わらないため、分割数分のレイテンシが増えること以外のパフォーマンス上のデメリットはない。

また、異なるノードとの通信の際に相手ノードの計算ユニットそのものを指定できるよう、ノード全体のIDを比較してオフチップのネットワークインターフェースへ転送する機能を実装した。これにより、ノード間での通信で送信先のアドレスに加えてネットワークインターフェースのアドレスを指定する必要がなくなり、パケット中の利用可能な空間に余裕が生まれる。

本実装では、拡張性と柔軟性を重視し、Chisel言語を用いて多くの部分をカスタマイズ可能とした。具体的には以下は全てカスタマイズ可能になっている。

- 2D メッシュトポロジにおける縦および横のルータ数
- フリット中で送信先アドレスがエンコードされるビット位置
- 計算ユニットがインターフェイスに渡すフリットのビット幅
- ルータ間を実際に送信するビット幅
- オフチップネットワークのインターフェイスのアドレス
- チャネルを時分割多重化する数
- ルータ内のキューの深さ

2.2.3.7　Compute Node(以下 CN)の調査結果

実際のアプリケーションでは、これまでに実装したハードウェアモジュールを組み合わせた計算ユニットを複数使用して一連の計算を行う。そのため、計算ユニットと、それをNetwork-on-Chip (NoC)で接続したハードウェアと、その動作検証用のソフトウェアを実装し、またFPGA上に実装した。

計算ユニット内には、イベント駆動モジュール、汎用コア、専用パイプライン、メモリが含まれる。これらを接続し、また動作検証用に設定からモックアップと切り替えられるようなハードウェアを実装した。また、これをシミュレーションするソフトウェアを実装した。ソフトウェアでは、商用のverilogシミュレータの他に、オープンソースのverilogシミ

ュレータであるverilatorでも検証を行えるようにした。通信に使用するパケットへデータをエンコードする関数と、verilogの機能であるC言語との相互利用をする関数はどのシミュレータを用いても利用できるよう実装し、またverilatorを用いた実装では通信ポートごとに独立して非同期な制御を行えるよう並列化を行った。

加えて、複数の計算ユニットでの動作検証を行えるよう、複数の計算ユニットと、それらを接続するNoC、そしてFPGA上でホストPCとの通信を行うインターフェイスと、FPGA間の通信を行うOff-Chip Networkへのインターフェイスを持つハードウェアを実装した。ここで、NoCによる接続とレイアウトを変更しやすいよう、NoCの設定と同時にレイアウトの設定を与えるようにし、Off-chip Networkのインターフェイスの数や計算ユニットの数を容易に変更できるようにした。また、この動作検証を行うソフトウェアも、計算ユニットと同様に実装した。

検証を行ったノード構成としては、CU×4、ホストインターフェース×1、ネットワークインターフェース×1を3×2のメッシュトポロジのチップ内ネットワーク（Network On Chip、以下NoC）に接続したNode（図 2.2.3.2）を構築し動作検証を行った。CU単体と同じ計算条件をホストインターフェース・NoC経由で1CUに書き込みCU単体評価と同じ計算結果がNoC・ホストインターフェースを経由して回収されることを回路シミュレーションで確認した。またCN回路をFPGAで実装し、回路シミュレーションと同じ計算結果が得られることを確認した。CUと同様に、FPGA実装結果に基づき回路の改良を行い、実装を効率化・タイミングの改善を図った。

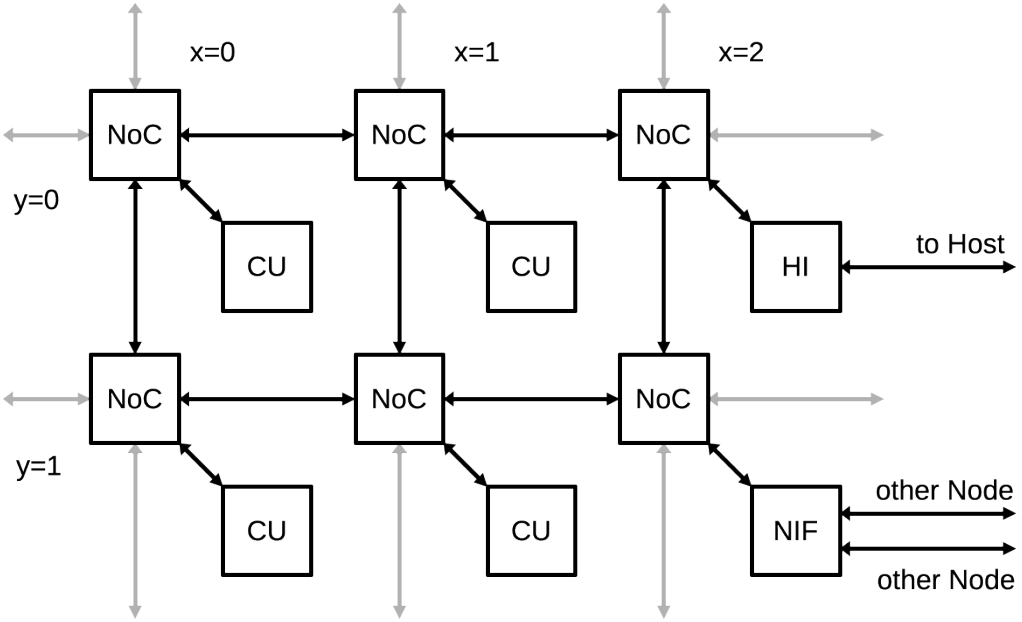


図 2.2.3.2　ComputeNode の構成図。CU: ComputeUnit, NoC: Network-on-Chip, NIF: Network Interface, HI: Host Interface.

2.2.3.8　複数ノードシステムの調査結果

まず複数ノード接続のため、オフチップネットワークの開発を行った。MDGRAPE-4Aのチップ間ネットワークを改良し、NoCで使われるデータパケットをチップ間で送受信できるように設計変更した。設計を回路シミュレーションとFPGA実装により動作検証した。

このオフチップネットワーク(NIF)を用い、4つのComputeNodeを相互接続した構成を作成し、評価を行った。4CNをNIFにより2×2メッシュ接続した状態（図 2.2.3.3）の回路シミュレーションを実施しホストインターフェース-NoC-NIF-NIF-NoC-CUという複数Nodeを経由してデータが書き込めること、逆の経路で計算結果が読み出せることを確認した。また、CU単体と同じ計算条件を上述の経路で書き込み、1CUで計算し同じ結果が得られることを回路シミュレーションで確認した。さらに、複数のCUを同時に動作させるテストも行った。4CNを4個のFPGAに実装し、実際の実行を行い、回路シミュレーションと同じ結果が得られることを確認した。

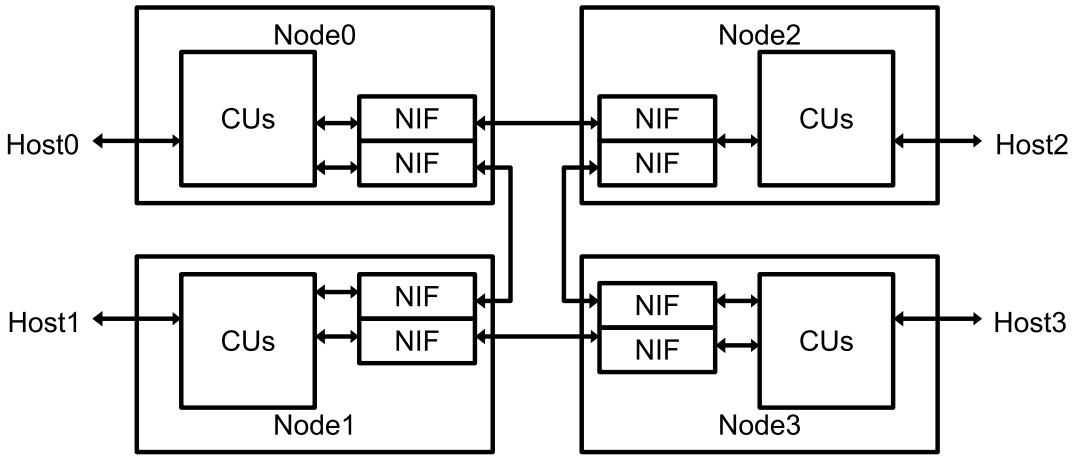


図 2.2.3.3　4 ノード構成のシステム図。

2.2.3.9　ASIC への実装調査結果

TSMC 7nm (N7)プロセスを用いたASICへの実装評価を行っている。CU1ノードの論理合成を行い、速度評価・電力評価を行った。本内容については、現時点では詳細は開示しない。

2.2.4　今後の課題と計画

今年度までに完成させたハードウェアの改良を図ると共に、より現実的なシナリオでの性能評価を行う。また、配置配線まで含めたより現実的なASIC実装を試行し、特に電力性能を評価する。また長距離力計算部の完成度を高め、アルゴリズムも含めた検討を行う。

2.3 アーキテクチャ調査研究サブグループ 2（富士通）

本年度は富岳後継機のアーキテクチャについて、初年度に実施した技術動向調査およびノード・システムアーキテクチャ検討の結果をベースに、より深い検討を実施した。

2.3.1 調査研究の概要

ノードアーキテクチャは三次元実装とチップレットを含む先進パッケージング技術を活用し、プロセッサ、アクセラレータ、高帯域メモリを統合するヘテロジニアスな密結合アーキテクチャを検討している。

初年度にはプロセッサでは富岳と同様にモノリシックなダイで構成するType-C、チップレットの再利用性を活用して設計コストを抑えたType-B、チップレットの低製造コストを活用して大型パッケージを実現するType-Aをモデル化した。アクセラレータでは密行列計算に特化した固定機能型のType-C、密行列計算に加えて計算科学で使用頻度の高い特定処理も加速するバランス型のType-B、原理的には幅広くアプリケーション一般を加速可能な再構成可能型のType-Aをモデル化した。また、アクセラレータを搭載しないType-0も選択可能とした。メモリではコストの低い安いモバイルメモリを活用するType-C、富岳と同様に高帯域メモリを使用するType-B、SRAMの三次元積層で非常に高い帯域を実現するType-Aをモデル化した。さらに、これらのモデルに技術動向調査のパラメータを適用し、ノードあたり製造コストと設計コストを富岳に対する比率で算出、比較した。

本年度はまずアクセラレータType-A/B/Cの面積、電力について初年度に実施した初期評価の精度を改善し、さらに富岳のHPL実行時電力を基準にノード消費電力をモデル化することで初年度各Typeのノード消費電力を評価した。コデザインについてはアプリケーション性能評価によるフィードバックは間に合わなかったが、学会発表やヒアリングを通じて初期検討モデルに対する多角的な評価を得た。具体的には5月のHPC研究会で初年度のプロセッサおよびメモリ評価結果、9月のHPC研究会で精度改善後のアクセラレータ初期評価結果とノード消費電力評価結果を発表し、さらに関係各所や有識者へのヒアリングを実施した。それらのフィードバックを得て、プロセッサおよびメモリTypeの追加やアクセラレータのアーキテクチャ具体化および改良を次の通り実施した。プロセッサはコア数を増やしたType-D、アクセラレータはHPC汎用性とAI性能を強化したType-D、先端プロセスを使用するType-E、ベクトル演算部を追加するType-F、メモリはホストメモリ追加した上でマルチソケット化するType-Dを追加した。本年度はこれらの新しいTypeを組みあわせた3つのノードアーキテクチャを11月末までに提案アーキテクチャとして取りまとめた。そして、12月の関係各所や有識者への再度のヒアリングを受けて、特に 1 つの組み合わせに対して、11月末の提案アーキテクチャに対して帯域2倍以上となるメモリの導入を検討し、これに対応するデータパス強化も実施した。また、この強化後のアーキテクチャに対して電力、面積の再評価も行った。

システムアーキテクチャはCPO使用とAOC使用を想定したインターコネクトの構成それぞれに対し、初年度に抽出したパラメータを適用して消費電力を見積もった。またデータセンターネットワークの技術動向から、OCSを使用する3D-Torusも検討視野に入れ、6D-Torusとジョブ充填率を比較するための構成案を策定した。さらにアプリケーションとのコデザインに向けて、集団通信時間を通信アルゴリズムごとに定式化した。ストレージシステムについてはSSD、HDD、CXLメモリの追加技術動向調査に加え、高速データ共有システムのユースケース調査を実施し、その結果から初年度に作成したモデルの改良と構成の具体化を行った。さらに各構成についてストレージシステム全体のコストとハードウェアの理論性能、容量を評価し、ストレージシステムのハードウェア構成を提案した。

2.4 アーキテクチャ研究 サブグループ 3(Intel)

本年度は富岳後継機のアーキテクチャについて、初年度に実施した技術動向調査およびノード・システムアーキテクチャ検討の結果をベースに、より深い検討を実施した。

2.4.1 研究概要

2.4.1.1 アーキテクチャに関する調査

私たちは、メモリ、ネットワーキング、構造の分野におけるいくつかの破壊的技術を組み合わせた提案に取り組んできた。この提案は「パス」と呼ばれる4つの選択肢を軸に展開され、これらは技術的な改善レベルと関連するリスクレベルの違いを表している。以下の表 2.4.1.1は、これらすべての技術をハイレベルに格付けしたものである（H-高、M-中、L-低）。高を破壊的技術、低を進化的技術と定義している。

表 2.4.1.1

構成技術	パス A	パス B1	パス B1.5	パス B2	パス C
メモリ	H	H	M	L	L
ネットワーキング	H	L	L	L	L
構造	H	M	M	M	L

5つのパスはすべて、スーパーノードあたりの生の演算性能は同じだが、SoP（System-on-Package）、スーパーノード、システムの技術ソリューションが異なると仮定している。

パスCは純粋に進化的なものであり、様々な破壊的技術の選択による改善の機会を明確にするための比較参考として提供される。

次世代コンピューティング・インフラの設計には、「京」や「富岳」、そして「Aurora」のような大規模システムの構築から得られた成功や教訓を基にしつつ、進化する計算ニーズやユーザ要件に対応する技術の進歩を考慮する必要がある。この目的を達成するためには、以下に述べるように、様々な領域にまたがる包括的な戦略が必要である。

2.4.1.1.1 アーキテクチャの革新

- パフォーマンスを最適化するために、CPU、GPU、その他のアクセラレータを組み合わせたヘテロジニアスアーキテクチャを実装する。
- 科学のための AI のトレンドと AI を取り入れた HPC ワークロードの進化に基づき、演算データ型の適切な組み合わせを提供する。
- 特殊なタスクに特化した量子コンピューティングやニューロモーフィック・コンピューティング、また従来のフォン・ノイマン・コンピュータを超える新しいアーキテクチャの実験的な活用を検討する。
- スケーラビリティと相互接続性
  - 増大するデータと計算需要に対応するためのスケーラビリティと、データ移動の削減/最適化を優先する。
  - ノード間の効率的な通信のための高度な相互接続ソリューションを開発する。
  - 低エネルギー、低レイテンシなデータ転送のための光インターコネクトを含める。
- エネルギー効率

- エネルギー効率の高い設計とシステムレベルの電力管理に重点を置き、電力消費の課題とエネルギーコストに対処する。
- 先進的な電力供給と冷却技術を取り入れると同時に、施設レベルで代替エネルギー源を模索する。
- 次世代アルゴリズムとハードウェアの協調最適化を行い、性能効率を向上させる。

2.4.1.1.2　信頼性、アクセス性、保守性(RAS)

- 堅牢なエラー検出・修正メカニズムを実装し、システム全体の遠隔測定・計測フレームワークによってサポートされる耐障害設計を行い、システムレベルのデータマイニング、診断、またデバッグを可能にする。
- システム設計とシステム管理に対する総合的なアプローチを提供し、スマートな予測／予防システム管理のための AI の使用について調査する。
- セキュリティおよびプライバシ
  - 機密データを保護するための強固なセキュリティ対策を導入する。
  - ハードウェアレベルのセキュリティ機能を統合し、新たな脅威から保護する。
  - 個人情報保護に関する規制や地域の基準に確実に準拠する。
- アクセシビリティおよびインターフェイス
  - 多様なユーザコミュニティのために、ユーザーフレンドリなインターフェイスを優先する。
  - 効果的なコラボレーション、データ共有、視覚化のためのツールを導入する。
  - インフラの効率的な利用を促進するプログラミングモデルと言語を開発する。
- ハイブリッド・クラウドの統合
  - ワークロードの需要に応じた柔軟なリソース割り当てと動的なスケーリングを可能にする。つまり、次世代システム管理ソフトウェアにクラウドベースの機能を組み込む。
  - ハイブリッド・クラウド環境とシームレスに統合するインフラを設計する。
  - 安全なクラウド統合を促進するために、セキュリティ対策を強化する。
- 持続可能性と環境への影響
  - インフラの設計と運用において環境への影響を考慮する。
  - 製造、運用、廃棄において持続可能な手法を導入する。
  - カーボンニュートラルまたは低炭素技術を探求する。
- 継続的フィードバックとモジュール性
  - 継続的なユーザフィードバックと要件収集の仕組みを確立する。
  - 技術革新の最前線に立ち続け、長期的かつ多世代にわたる技術パイプラインを提供するために、研究コミュニティとの協力を促進する。
  - システムの部分的なアップグレードやより長期間にわたり全体的なシステムの再利用を可能にするためにシステムのモジュール性を提供する。

このような配慮を開発プロセスに取り入れることで、次世代コンピューティング・インフラストラクチャは、進化する課題に対応し、新たなテクノロジを受け入れ、刻々と変化する計算科学技術の状況におけるユーザの多様なニーズに応えることが可能となる。

2.4.1.2　コンパイラおよびソフトウェア環境に関する調査

oneAPIコンパイラ（SYCLやOpenMPのようなプログラミングモデルのサポートを含む）と、より高レベルのオープンソースライブラリ（oneAPI Deep Neural Network Libraryなど）およびオープンソースプロジェクト

（Intel® Neural CompressorやSYCLomaticなど）を使用し、CPUやアクセラレータ間で共通のプログラミングモデルを提供することを提案した。

oneAPIが、さまざまなベンダの多種多様なハードウェアアーキテクチャにわたって、パフォーマンス、移植性、生産性の高いプログラミングを実現する魅力的なプログラミングモデルであることは、すでにいくつかのケーススタディで実証されている。ここでは網羅するリストを提供しないが、それに関するリストはIWOCL & SYCLcon (https://www.iwocl.org/) やP3HPC (https://p3hpc.org) などのワークショップのウェブサイトに掲載されている。特に興味深いのはEstebanらによる論文「A Performance-Portable SYCL Implementation of CRK-HACC for Exascale」（P3HPC'23に掲載予定、 https://arxiv.org/abs/2310.16122v1）であり、エネルギー省のAurora、FrontierおよびPolarisマシン（それぞれIntel、AMD、NVIDIAのGPUを採用）においてSYCLアプリケーションが高レベルの性能を達成できることを実証している。

図 2.4.1.2は、oneAPI プログラミングモデルのオプションのうち、富岳NEXT に最も関連すると思われるサブセットを示している。サポートされているプログラミング言語は複数あるが（図の左側に記載）、最終的にはすべて単一のソフトウェアスタックと統一されたランタイムによって駆動される。この設計は、これらのモデル間の相互運用性を可能にし、高レベルのOpenMPディレクティブと低レベルのSYCLカーネルを（クリティカルなホットスポットを高速化するために）混ぜて使用するアプリケーションの例が既に幾つか存在する。また、oneAPI ソフトウェアスタックに代わるインターフェイスを提供するコミュニティ・プロジェクト（Kokkos、RAJA、OCCA、chipStar、YAKL、複数のドメイン固有言語など）も複数存在する。ただし、これらのコミュニティ・プロジェクトはインテル製品ではないため、富岳NEXT のタイムフレームにおける可用性や適合性については保証できない。

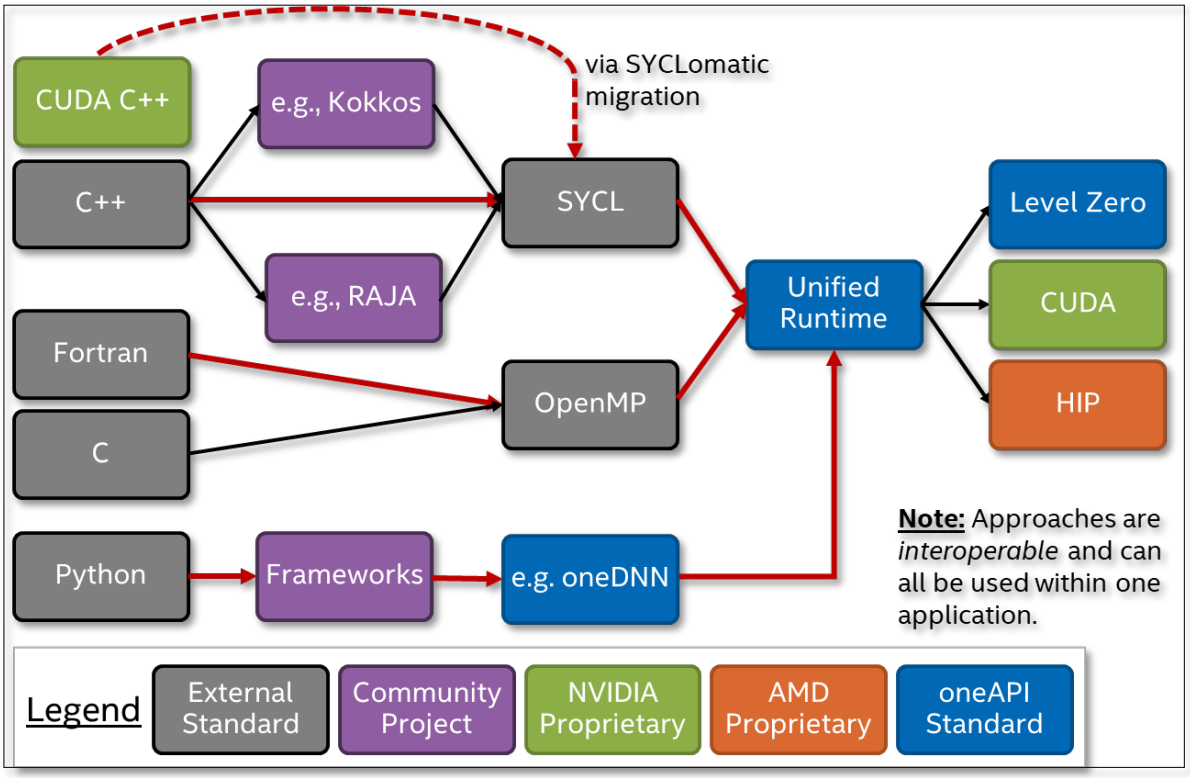


図 2.4.1.1　oneAPI で有効なプログラミング・オプション。
赤線で示されているのが新しいコードの推奨パスである。

我々が推奨する新しいコードの開発経路は、図 2.4.1.2中の赤線で示している。利用可能なフレームワーク（例えば、機械学習や人工知能にPyTorchを使用）を活用することで、関連する領域により近い抽象度の高

いアプリケーションを開発できるようになり、プログラマの生産性が最大化される。C++とSYCLの併用は、新しいライブラリの開発や、大規模な科学ソフトウェアパッケージの開発に適しており、開発者は、問題がどのように基礎となるハードウェアリソースにマッピングされるかを正確に制御する必要がある。FortranとCで書かれたレガシーアプリケーションは、OpenMPによって提供されるオフロードコンストラクトを介して、oneAPIスタックによって完全にサポートされるが、これらの言語で新しいコードの開発を始めることは推奨されない。

2.4.1.2.1 LLVM への SYCL サポートのアップストリーム

インテルは、oneAPI DPC++コンパイラに実装されているSYCLサポートをLLVMプロジェクトにアップストリームするための作業を積極的に行っている。これにより、すべてのLLVMベースのコンパイラ（clang++など）がSYCLソースコードを解析してコンパイルできるようになり、SYCLのコミュニティ実装（つまり、特定のベンダに縛られない実装）によって、より広範なターゲットデバイスがサポートされるため、SYCLの採用が加速することが期待される。アップストリームへの取り組みの概要は、https://discourse.llvm.org/t/rfc-add-full-support-for-the-sycl-programming-model/ に投稿されている。

2.4.1.2.2 The Unified Acceleration (UXL) ファウンデーション

Linux Foundationは最近、Unified Acceleration (UXL) Foundation (https://uxlfoundation.org/) の設立を発表した。設立メンバには、Arm、富士通、Google Cloud、Imagination、Intel、Qualcomm、Samsungが名を連ねている。UXL財団は、多くの点でoneAPIイニシアチブを進化させたものであり、マルチベンダ運営委員会への移行は、oneAPIが複数のベンダの複数のアーキテクチャをターゲットとする能力を継続的に向上させるのに役立つ。この変更は、Khronos Group Inc.が開発した業界標準であるSYCLプログラミング言語には影響を与えないことに留意することが重要である。むしろ、UXL Foundationの役割は、標準的なライブラリやフレームワーク（例えば、ディープニューラルネットワーク、データ解析、一般的な並列パターンなど）の幅広いアクセラレータエコシステムの開発を推進することであり、これらはすべてSYCLプログラミング言語を基盤としている。

2.4.1.2.3 理研が有するコードにおいて想定される変更

この研究のためにインテルに提供された理研のコードのほとんどは一部を除き、現在GPUに対応していないか開発中である。

アプリケーションの移植性を高めるという我々の目標を達成するためには、おそらくこれらの実装をOpenMPかSYCLに移行する必要がある。

SYCLとOpenMPは、CUDAとOpenACC（それぞれ）とは異なるセマンティクスを持つ場合があり、これは一方から他方への移行プロセスを複雑にする可能性がある。このセクションで説明するツールは、可能な限り移行プロセスを支援するように設計されている。実際には、開発者が手動で介入することなく、コードベースの90～95%を移行できると期待している。ただし、マイグレーションによって機能するコードが生成される場合でも、対象となる特定のアーキテクチャの特性に合わせてチューニングすることによってパフォーマンス（移植性）を向上させるためには、開発者による手作業が推奨されることに注意が必要である。

CUDAからSYCLへの移行は、LLVMをベースにした成熟したオープンソースプロジェクトであるSYCLomatic (https://github.com/oneapi-src/SYCLomatic)を使用して実行できる。OpenACC から OpenMP への移行は、Intel® Application Migration Tool for OpenACC to OpenMP (https://github.com/intel/intel-application-migration-tool-for-openacc-to-openmp) でも可能である。

これらの移行ツールは現在開発中であり、本研究の一環として行った取り組みにより、いくつかの改善点が明らかになっている。

インテル® ソフトウェアスタックは、インテル® オフロード・アドバイザ® を使用して、オフロードするのに適した候補を判断するのに役立つ。このツールはCPU コード上で実行することができ、さまざまな分析を通じて、オフロードすべきコードをハイライトし、可能な限り最良のスレッド占有率を得ることができる。このツールは、SYCL や C++ 経由でコードを移植する場合にも使用できる。

2.4.1.3 ストレージおよびファイルシステム

2.4.1.3.1 要約および重要事項

本章の目的は、インテルが提案する富岳NEXTアーキテクチャをサポートするのに十分な容量、性能、ソフトウェア機能を持つプライマリストレージシステムの可能な構成を理研に提供することである。

ストレージ分野におけるインテルからの提案は、Distributed Asynchronous Object Storage (DAOS)と呼ばれるエクサスケール・フラッシュ・ストレージのシステムである。DAOSは、既に例えばALCFのAuroraなどにおいて、他の高性能ストレージソリューションよりも桁違いに優れたスケーラビリティと性能レベルがある事を実証している。我々は、DAOSが富岳NEXTのプライマリストレージシステムにとって最適であると考えている。

コストの観点からすると、HDDで構成された10倍程度の容量を持つプライマリストレージシステムを持つことが好ましいと言える。DAOSは伝統的なLinux block I/O storage stackをバイパスする完全フラッシュストレージシステムであるためHDDはサポートしない。インテルのDAOSチームは、HDDベースのシステムを提供する富岳NEXTのインテグレータやHPC/AIのソリューションパートナー(富士通やDDNなど)と協力し、DAOSを費用対効果の高いシステムにしたいと考えている。本件に関しては、本レポートの範囲外である。

現在のアーキテクチャ・リサーチ・グループからの要件には、ストレージに関する要求やベンチマークは含まれていない。そこで我々が提案するアーキテクチャに適したストレージシステムのハードウェア構成を得るために、本レポートでは「チェックポイント」に着眼し以下の原則でストレージのサイジングを行った。

- ストレージのサイジングは、システムのトータルメモリ容量で決まる。
- パフォーマンスのサイジングは、全システムのメモリがストレージに書かれる許容可能な時間により決まる。
- キャパシティのサイジングはプライマリストレージに何個のメモリコピーが存在しなくてはならないかにより決まる。

この方法論は、建部教授が率いる「システムソフトウェアとライブラリ」研究グループのIO・ストレージ・ファイルシステムサブグループでの議論とも一致している。我々は、今回提案するDAOSが、HPCやAI分野において、古典的なHPC checkpointsを凌駕して適応可能であるという事をここに強調する。多くのAIワークロードでは、多数の小さなファイルの読み込みが重要であり、DAOSはこれらのワークロードに関しても適している。ストレージのサイジングは、より詳細なストレージ要求が決定された際にも修正可能である。

コストを最適化するために「システムソフトウェアとライブラリ」研究グループはプライマリストレージシステムを全てフラッシュシステムとして実装すべきか、あるいはパフォーマンス重視の小容量のSSDと容量重視のHDDを組み合わせた2層構造とすべきかについても議論している。我々は、プライマリストレージシステムは完全にフラッシュシステムにするべきであり、HDDは広域ストレージシステムに限定すべきであると考えている。我々のコスト最適化へのアプローチは、高レベルのストレージ要件に基づき各DAOSストレージエンジン内で異なる品質のNVMe SSD(物理メディアタイプ、耐久性、性能)を組み合わせて使用することである。システムアーキテクチャおよびユーザから見ると、我々のソリューションは、各DAOSストレージエンジン内の異なる特性のNVMeメディアを透過的に管理する内部配置ポリシーを備えた単一のNVMeストレージシステムに見える。

これらの境界条件から、本レポートでは富岳NEXT向けのDAOSストレージソリューションのために可能なハードウェア構成の概要を示し、既存のDAOS実装と比較する。またHPCネットワークと全体的なスケーラビリティに関連するストレージソフトウェアアーキテクチャのいくつかの重要な側面を強調する。ただし、ここでストレージシステムの機能とコストは、性能(例えば、チェックポイントの実行時間)と容量に対する(想定される)要件を変更することにより計算システムとは独立して拡張できることに留意されたい。

本章は以下の様に構成される。

- 2.4.1.4.2 章では計算システムが如何にストレージサイジングに影響を与えるかに関して解説する。
- 2.4.1.4.3 章では DAOS ストレージスタックの全体像を解説する。
- 2.4.1.4.4 および 2.4.1.4.5 章ではインテルのアーキテクチャに対してバランスの取れたストレージのハードウェアデザインを解説する。
- 2.4.1.4.6 章ではストレージソフト側の「教訓」を指摘し将来のコデザインに対して解説する。
- 最後に幾つかの参考文献を紹介する。

2.4.1.3.2 アーキテクチャにおけるストレージのサイジングへの影響

インテルは演算性能目標を達成するために、異なる性能と技術的リスクを伴ういくつかの選択肢を検討している。ストレージの観点から見ると、これらの選択肢は主にシステムメモリの合計サイズの違いに起因している。

- より現実的なストレージは、PCIe gen6 NVMe SSD と PCIe gen6 ネットワーキング(リンクあたり 800 Gbps)を備えた次世代ストレージサーバを使用する既存のテクノロジをそのまま進化させたストレージシステムである。このストレージソリューションは技術的リスクが低い。
- より先進的な選択肢とそれに関連するシステムメモリサイズについては、PCIe gen7 ベースのストレージサーバをファブリックにアタッチすることを想定しているが、PCIe gen7 NVMe デバイスは 2030 年には利用できない可能性が高い。必要な帯域幅を提供するためには PCIe gen6 の NVMe 数を比例して増やす必要がある。これによってスケーリング要件は増加するが、ストレージソリューションの技術リスクは依然として低～中程度となる。

これらのストレージソリューションパスについては、本レポートの残りの部分で個別に説明する。

2.4.1.3.3 Distributed Asynchronous Object Storage (DAOS) 概観

インテルはDAOSを富岳NEXT向けの主ストレージシステムとして提案する。DAOS[1-5]はオープンソースプロジェクトで、中心となっている開発チームはインテルであるが、コミュニティ(アカデミックや産業界含む)からのx86_64やARM64への寄与も非常に大きい。このオープン性が、プロプリエイタリなストレージソリューションに対する戦略的優位である。国際的な協力が可能になるだけでなく、DAOSをベースにしたストレージソリューションの開発の可能性も広がる。大規模なHPC/AIや特定の科学的領域のためにDAOSの機能を共同設計することは、当初からDAOS開発チームの基本理念であった。例えば今日のDAOSソフトウェア機能の多くは、Auroraエクサスケールシステムの一環として、アルゴンヌ国立研究所(ANL)とコデザインされてきた。富岳NEXTにストレージソリューションを最適に統合するために、新機能の開発が有益となる分野がさらに特定されることを期待している。

DAOSプロジェクトのガバナンスに関して、インテルと他の主要なDAOS関係者が近年、Linux Foundationの傘下にDAOS foundationを設立したことを発表している。詳しくは以下のURLを参照されたい。https://www.linuxfoundation.org/press/daos-foundation-launches-to-broaden-governance-of-distributed-asynchronous-object-storage

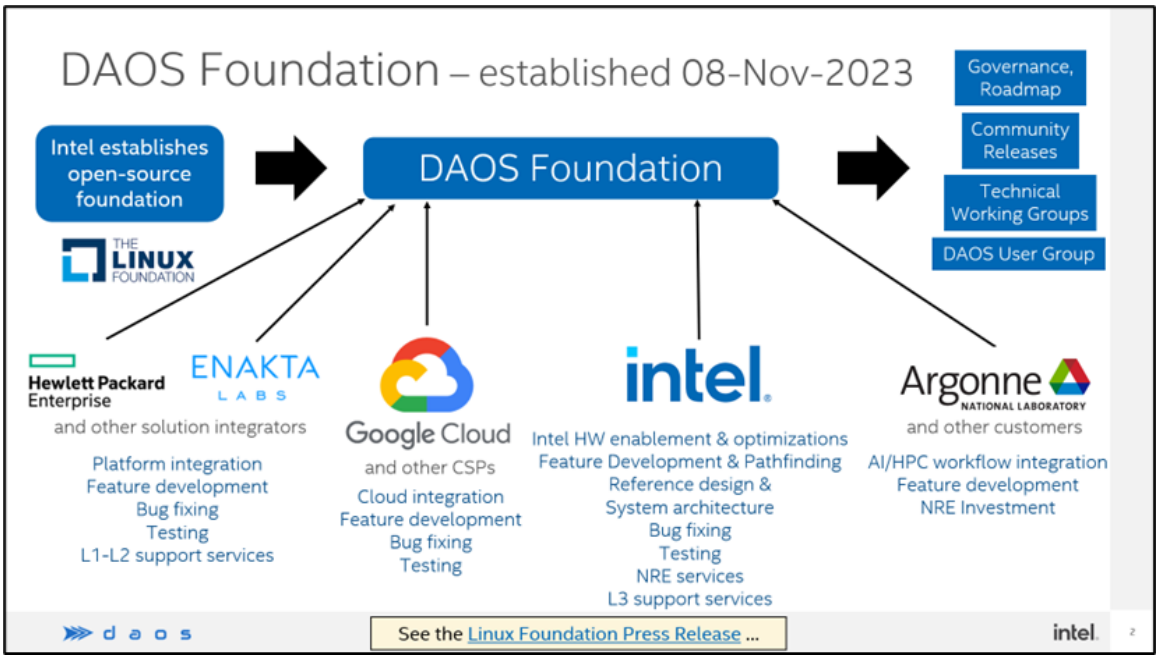


図 2.4.1.2　DAOS Foundation

DAOSのストレージ構成は、古典的なHPCストレージシステムのそれとは大きく異なる。この特徴は、エクサスケールかあるいはそれ以上のスケールのスーパーコンピュータに対して理想的であると言える。

- 完全に分散したメタデータ。専用のメタデータサーバを必要とすることはなく、メタデータの容量と性能はシステムサイズに応じてスケールする。この設計は、フルシステムスケールで DAOS ストレージプールを展開する柔軟性を提供するだけでなく、ワークロード分離のためにストレージハードウェアの別々のサブセット上でストレージをより小さなプールに簡単に「分割」することを可能にする。より高度な DAOS QoS については、2.4.1.4.7.2 章を参照されたい。この機能は既存の並列ファイルシステムには存在しておらず、これらの環境では「サービスの品質」を保証することが難しくなっている。
- DAOS 統一モデルは、独自の Versioning Object Store (VOS)を使用する。これは DAOS オブジェクト API にマッピング可能な POSIX および他のドメイン固有のデータモデルの両方に対して、大きなパフォーマンス上の利点を提供する。DAOS ストレージアーキテクチャの機能をフルに活用するために、ユーザは既に DAOS バックエンドを持つ既存のフレームワーク（MPI-IO、HDF5、Tensorflow-I/O、PyDAOS など）を使用することが推奨される。主なスーパーコンピュータ上の主要プロジェクトでは、アプリケーション固有の DAOS コンテナタイプを通じて、ドメイン固有のデータセット管理の DAOS コンセプトを採用することも有益であり、パフォーマンスを継続的に桁違いに向上させることができる。
- DAOS は完全にユーザ空間で動作し、かつ従来の OS ストレージスタックをバイパスし、最新の NVMe および SCM ストレージデバイスのハードウェア性能をフルに活用する。また、DAOS ファイルシステム (DFS) API を直接使用できないレガシーな POSIX ベースのアプリケーションのために、I/O インターセプションライブラリを使用することが可能である。これは、完全なユーザ空間 POSIX サポートを提供し、メタデータ操作を含むすべての POSIX コールをインターセプトし、userfaultfd を介して mmap のサポートも提供する。

上記のソフトウェア機能(これらはすでに製品機能として確立しているか、または開発途上にある)に加え、高性能ストレージスタックにコンピューテーショナルストレージ機能を導入することが極めて重要になると考える。ストレージサーバからクライアントに転送されるデータ量を削減するためには、サーバ側でデータのフィルタリングや前処理を可能にする技術を検討する必要がある。我々はすでに、サーバーサイドの名前空間トラバーサル

（POSIX find）のプロトタイプ実装を披露しており、大規模なスケールアウトストレージシステムにおけるこのアプローチの実現可能性を実証している。

インテルは近年、Optane Persistent Memory製品ラインの停止を発表した。オリジナルのDAOSソフトウェアスタックは、ストレージクラスメモリ(SCM)レイヤとしてOptane PMemを使用していた。最新のDAOS 2.4リリースには「MD-on-SSD」として知られる代替コードパスが含まれており、NVMe SSD上のDRAMとwrite-ahead-logで動作することができる[6,7]。図 2.4.1.4に示すように、DAOSアーキテクチャ全体に変更はなく、性能テストでは、この新しいコードパスはPMemベースのDAOSシステム[8]と非常によく似た性能であることが示されている。

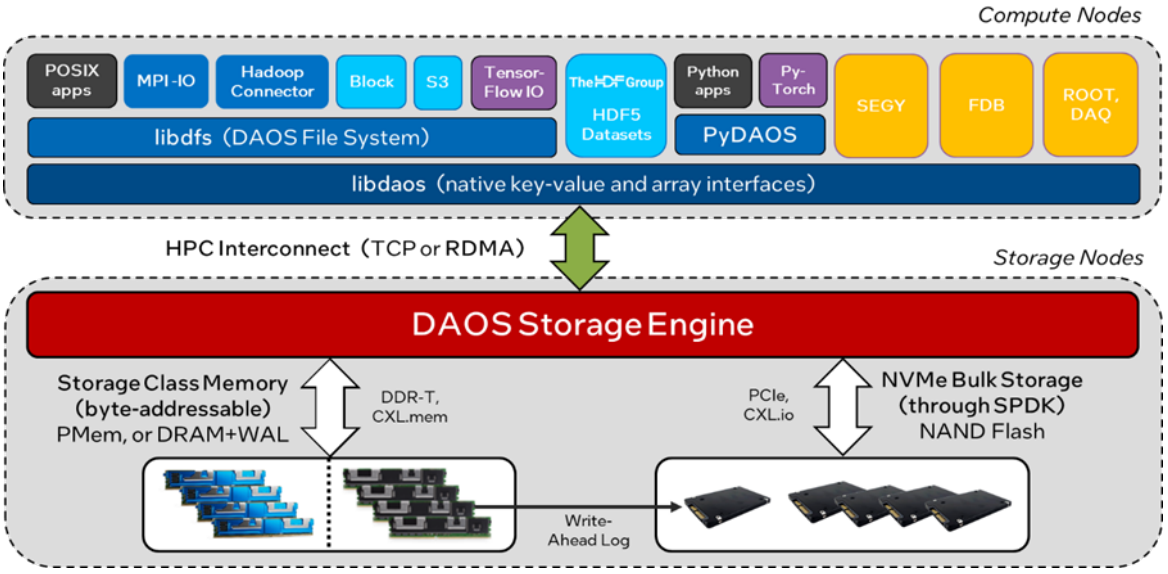
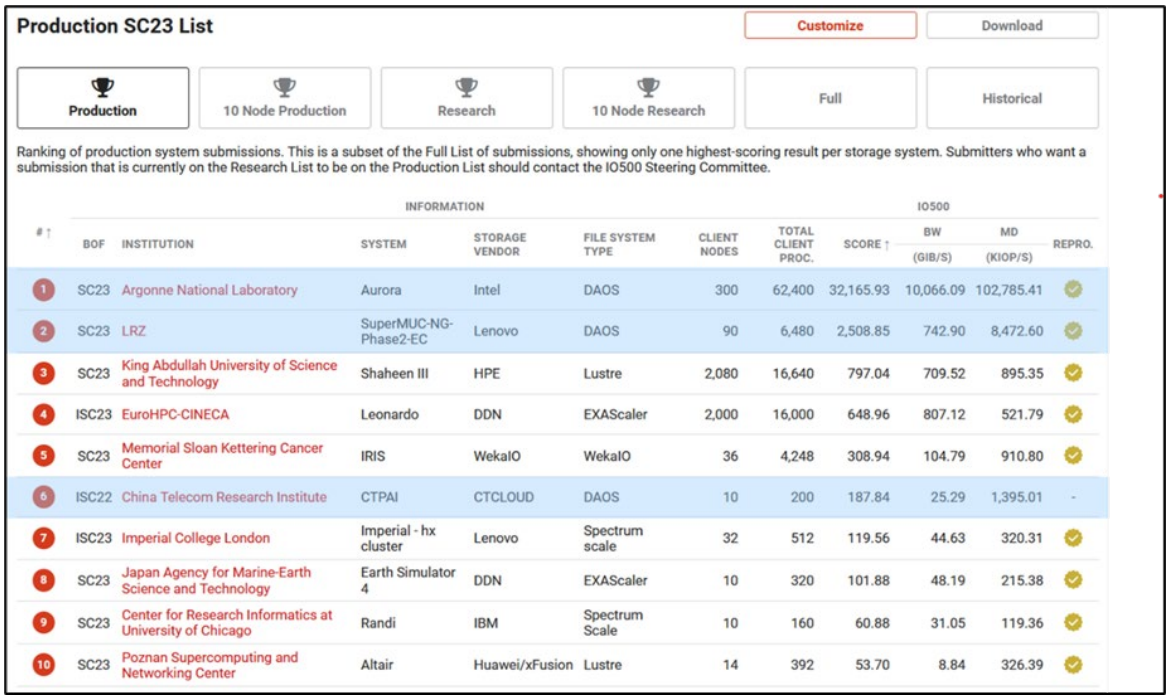


図 2.4.1.3　DAOS アーキテクチャの概観

2.4.1.3.4　DAOS パフォーマンスの証明

DAOSは、LustreやGPFSのような従来のHPCファイルシステムのパフォーマンス上の制限に対処するために作成された。そのVOSデータ構造は、バイトアドレッシング可能なストレージクラスメモリ(永続メモリかDRAM + NVMeへのwrite-ahead-logのいずれか)に保持され、NVMeディスクはユーザスペースではバルクストレージとしてアクセスされる。DAOS VOSの設計は、大規模な従来のPOSIXファイルシステムで性能問題を引き起こすロック、同期、false sharingの問題に悩まされることなく、任意のサイズと任意のアライメントの非破壊的書き込みを可能にする。一方、DAOSは、すべての高性能ストレージシステムにおいて、明確なパフォーマンスリーダとしての地位を確立している。図 2.4.1.5は、最新の IO500-SC23 ストレージシステムのトップ 10 を示しており [9]、現在 Argonne と LRZ DAOS ストレージシステムが 1 位と 2 位を占めている。



Production SC23 List

Customize | Download

Production | 10 Node Production | Research | 10 Node Research | Full | Historical

Ranking of production system submissions. This is a subset of the Full List of submissions, showing only one highest-scoring result per storage system. Submitters who want a submission that is currently on the Research List to be on the Production List should contact the IO500 Steering Committee.

#		INFORMATION							IO500		
	BOF	INSTITUTION	SYSTEM	STORAGE VENDOR	FILE SYSTEM TYPE	CLIENT NODES	TOTAL CLIENT PROC.	SCORE ↑	BW (GIB/S)	MD (KIOP/S)	REPRO.
1	SC23	Argonne National Laboratory	Aurora	Intel	DAOS	300	62,400	32,165.93	10,066.09	102,785.41	
2	SC23	LRZ	SuperMUC-NG-Phase2-EC	Lenovo	DAOS	90	6,480	2,508.85	742.90	8,472.60	
3	SC23	King Abdullah University of Science and Technology	Shaheen III	HPE	Lustre	2,080	16,640	797.04	709.52	895.35	
4	ISC23	EuroHPC-CINECA	Leonardo	DDN	EXAScaler	2,000	16,000	648.96	807.12	521.79	
5	SC23	Memorial Sloan Kettering Cancer Center	IRIS	WekaIO	WekaIO	36	4,248	308.94	104.79	910.80	
6	ISC22	China Telecom Research Institute	CTPAI	CTCLOUD	DAOS	10	200	187.84	25.29	1,395.01	-
7	ISC23	Imperial College London	Imperial - hx cluster	Lenovo	Spectrum scale	32	512	119.56	44.63	320.31	
8	SC23	Japan Agency for Marine-Earth Science and Technology	Earth Simulator 4	DDN	EXAScaler	10	320	101.88	48.19	215.38	
9	SC23	Center for Research Informatics at University of Chicago	Randi	IBM	Spectrum Scale	10	160	60.88	31.05	119.36	
10	SC23	Poznan Supercomputing and Networking Center	Altair	Huawei/xFusion	Lustre	14	392	53.70	8.84	326.39	

図 2.4.1.4　DAOS は IO500-SC23 のリストをリードしている

以下は、DAOSの特性の中でIO500やその他のベンチマーク結果を解釈する際に特に注目すべきものである。

- DAOS はデータとメタデータのサーバまたはターゲットを分離していない。すべての DAOS ストレージエンジンは、データとメタデータの両方を保持し、システムに DAOS ストレージエンジンを追加した場合、両者のパフォーマンスは完全にスケールする。これは、多くの場合専用のメタデータサーバまたはターゲットを持ち、利用可能なメタデータサーバまたはターゲット間でメタデータ操作を並列化するための比較的粗い粒度のメカニズムを持つ他のストレージソリューションとは大きく異なる。
- DAOS では、「単一の共有ファイル」と「プロセスごとのファイル」のアクセスパターンの間に性能差はない。多くのタスクから 1 つの大きなファイルのセクションにアクセスする(読み取りまたは書き込み)ために必要なロックやトークン管理がないため、これら 2 つのモードの性能は実質的に区別が不可能である。
- VOS ツリーに書き込みを非破壊で記録できるため、DAOS は他のストレージシステムと比較した場合、非常に高い IOR-Hard 帯域幅でも優れている。タスクあたりの帯域幅は IOR-Easy よりも低いが、これはメガバイトサイズの I/O を実行するよりも 47008 バイトの I/O を実行する方が効率が悪いためである。しかし、十分な数の MPI タスクがあれば、DAOS IOR-Hard の帯域幅の合計は、ストレージメディアのピーク帯域幅に近づく。これは、IOR-Hard が IOR-Easy より何桁も遅い他のストレージソリューションでは通常当てはまらない。
- DAOS のメタデータ操作は非常に高速である。Aurora マシンは、1024 台の DAOS サーバのうち 642 台と比較的少数のクライアントのみを使用して、MDTEST で平均 1 億回/秒以上のメタデータ操作が可能である事を実証済みである。これには利用可能である事を保証するために複製されたメタデータが含まれ、すべてのメタデータはストレージに永続化される。

図 2.4.1.6は、AuroraとLRZのハードウェア構成の詳細を示している。両システムとも、Ice Lake CPU、200Gbps NIC×2、Intel Optane 200 Series Persistent Memory、PCIe gen4 NVMe SSD(サイズと数量が異なる)を搭載したデュアルソケットDAOSサーバを使用している。両システムの計算ノードはSapphire Rapids世代のCPUとIntel Xe Ponte Veccio GPUをベースにしている。両システムは、

計算ノードあたりのネットワークポート数が異なる。AuroraはHPE SlingshotファブリックをLRZはNVIDIA HDR InfiniBandファブリックを使用している。

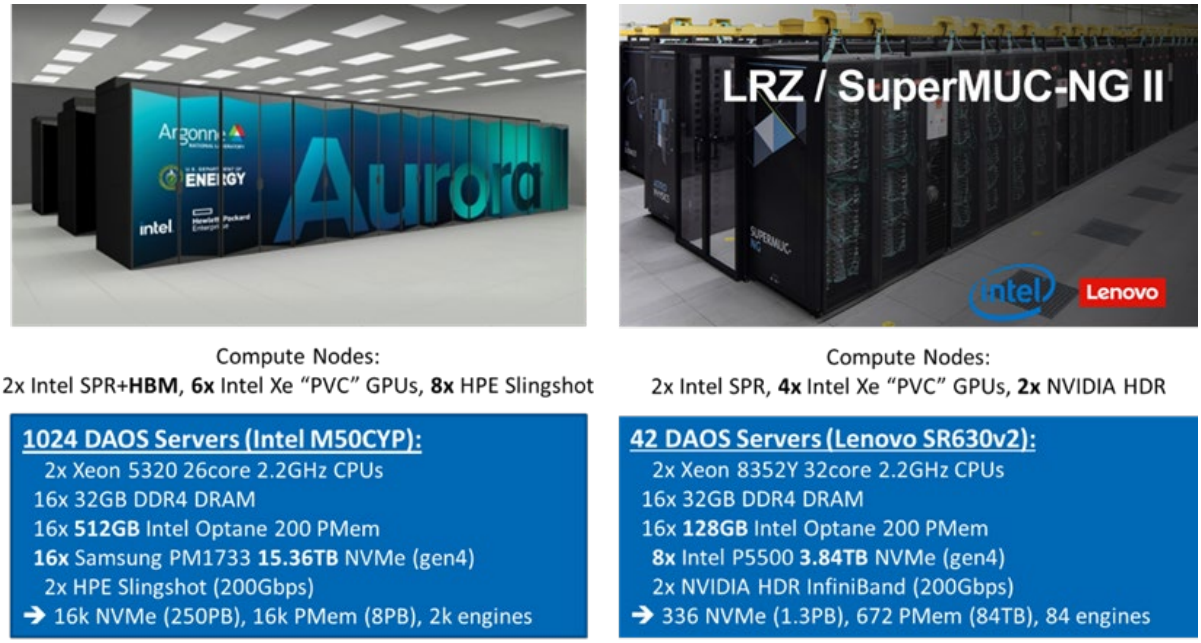


図 2.4.1.5　ALCF Auroraと LRZ SuperMUC-NG2 の設定詳細

2台のマシンのIO500スコアが、IO500-SC23ベンチマークに使用されたサーバ数の比率にほぼ比例していることは、DAOSスケールアウトソリューションのスケーラビリティを証明している。 Aurora は642台のサーバを使用し、LRZ は42台のサーバを使用した。IO500総合スコアの比率は12.8:1、帯域幅は13.5:1、メタデータは12.1:1である。完璧なスケーリング比からのわずかなずれは、HPCファブリックの違い(InfiniBand Fat treeとSlingshot Dragonflyの違い)、およびクライアントとサーバの比率の違いによるものと思われる。

参考文献

1. Liang, Z., Lombardi, J., Chaarawi, M., Hennecke, M.: DAOS: a scale-out high performance storage stack for storage class memory. In: Panda, D.K. (ed.) SCFA 2020. LNCS, vol. 12082, pp. 40–54. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-48842-0_3
2. Liang, Z., Fan, Y., Wang, D., Lombardi, J.: Distributed transaction and self-healing system of DAOS. In: Nichols, J et al. (eds.) SMC 2020. CCIS, vol. 1315, pp. 334–348. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-63393-6_22
3. Scot Breitenfeld, M., et al.: DAOS for extreme-scale systems in scientific applications (2017). https://arxiv.org/pdf/1712.00423.pdf
4. Hennecke M. (2023). Understanding DAOS Storage Performance Scalability. Proceedings of the HPC Asia 2023 Workshops. https://doi.org/10.1145/3581576.3581577
5. Hennecke M, Matsuda M and Nakao M. (2023). Evaluating DAOS Storage on ARM64 Clients. Proceedings of the HPC Asia 2023 Workshops. https://doi.org/10.1145/3581576.3581616
6. Lombardi, J., et al.: Metadata on SSDs design documentation (2022). https://daosio.atlassian.net/wiki/spaces/DC/pages/11196923911/Metadata+on+SSDs
7. Niu, Y.: WAL detailed design (2022). https://daosio.atlassian.net/wiki/spaces/DC/pages/11215339529/WAL+Detailed+Design
8. Hennecke, M. et al. (2023). DAOS Beyond Persistent Memory: Architecture and Initial Performance Results. In: Bienz, A., Weiland, M., Baboulin, M., Kruse, C. (eds) High Performance Computing. ISC High Performance 2023. Lecture Notes in Computer Science, vol 13999. Springer, Cham. https://doi.org/10.1007/978-3-031-40843-4_26
9. IO500-SC23 Production System List. https://io500.org/list/sc23/production

2.5 アーキテクチャ調査研究サブグループ 4 (AMD)

2.5.1 調査研究の概要

AMDは、次世代計算基盤の計算ノードのアーキテクチャの調査および研究において、特別な立場にあるといえる。AMD EPYC™ プロセッサ、AMD Instinct™ アクセラレータ―、ROCm™ オープン・ソフトウェア・プラットフォームを採用した「Frontier」は、2024年3月時点で世界最速のスーパーコンピュータであり、1 Exaflopsの壁を越えた唯一つのシステム[1]である。この比類のない性能に加えて、AMDは電力効率に優れた設計でも実績があり、Green500[2]では上位10位のうち7つのシステムにおいて、AMDのCPUとアクセレレータが採用されている。

こうしたハイパフォーマンス・コンピューティング (HPC) に関する専門知識を活用し、AMDは富岳NEXTの計算ノードのアーキテクチャ候補技術を提案した。半導体のプロセス技術の進歩が鈍化し、シリコンのコストが上昇している中、パッケージングおよび積層技術はますます重要になってきている。提案されたアーキテクチャでは演算器、メモリ、通信の密結合により、データ移動のオーバーヘッドが削減され、電力あたりの性能を向上させることができる。

調査研究の結果は非常に有望なものであった。AMDは提案アーキテクチャにおける理論ピーク性能、電力、電力効率の予測を実施した。さらに、理化学研究所で使用されているHPCアプリケーションや、急速に需要が増しているAIアプリケーションの性能も予測した。

現行の富岳に対して性能や効率が大幅に向上するこの提案アーキテクチャは、「研究デジタルトランスフォーメーション」とSociety 5.0の実現に不可欠である。

計算ノードの提案に加えて、AMDの最新技術を取り入れた新しいアーキテクチャの検討が、理化学研究所と共同で進められた。検討にあたっては、富岳NEXTの目指す方向性や求める要件を理解することが、より適したアーキテクチャの提案に非常に有効であった。

AMDは富岳NEXTプログラムの初期段階の調査研究を成功させ、結果としてさらに研究開発が必要な領域が明らかになった。これにはプロセッサ、アクセラレータ、パッケージング及び積層技術、メモリと通信技術、そしてソフトウェアが含まれる。ハードウェア、ソフトウェア、システムについては、初期段階からの共同設計が不可欠である。また、急速に進化するコンピューティング環境のリスクを低減する一つの方法は、5年ごとなどの短期間でシステムの更新を行うことである。これにより将来のハードウェアの構成要素やアプリケーション、システムソフトウェアを、継続して研究開発することが可能となる。

AMDのハードウェアとソフトウェアに関する多岐にわたる専門知識と、これまでに達成した革新的な成果、そして様々な機関との協力経験が、富岳NEXTの成功に重要な貢献をすると期待している。

[1] 62nd TOP500 List. https://www.top500.org/lists/top500/2023/11/.

[2] 22nd GREEN500 List. https://www.top500.org/lists/green500/2023/11/.

revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED ‘AS IS.” AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Third-party content is licensed to you directly by the third party that owns the content and is not licensed to you by AMD. ALL LINKED THIRD-PARTY CONTENT IS PROVIDED “AS IS” WITHOUT A WARRANTY OF ANY KIND. USE OF SUCH THIRD-PARTY CONTENT IS DONE AT YOUR SOLE DISCRETION AND UNDER NO CIRCUMSTANCES WILL AMD BE LIABLE TO YOU FOR ANY THIRD-PARTY CONTENT. YOU ASSUME ALL RISK AND ARE SOLELY RESPONSIBLE FOR ANY DAMAGES THAT MAY ARISE FROM YOUR USE OF THIRD-PARTY CONTENT.

Trademark Attribution Statement

AMD, the AMD Arrow logo, AMD EPYC, AMD Instinct, AMD Infinity Fabric, AMD ROCm, AMD 3D V-Cache, Xilinx, Pensando, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

2.6 アーキテクチャ調査研究サブグループ 5（NVIDIA）

2.6.1 調査研究の概要

＜提案・調査の方針＞

富岳NEXTのフィジビリティスタディにおいて、NVIDIAは2022年度に引き続き、本年度も以下を提案コンセプトとし、調査研究を進めた。

(1) 多様化するワークロードや予測不能な地球環境、社会環境の変化に対応するため、NVIDIAが考える将来のアーキテクチャや技術を基に、 GPU-CPU-DPU を統合した柔軟性に優れたデータセンタの構築を提案する。

(2) 2028年世代を見据え、優先されるべき特定分野のアプリケーションやアルゴリズムに最適なハードウェアやソフトウェアをどう実装すべきか、理研とのコデザインの中で議論する。ただし予測不能な数年先に向けたシステムの設計となるため、議論は調査研究以降も最終段階まで継続する。

(3) 限られたエネルギー資源を効率的に使い、必要とされるアプリケーションの性能を最大化することが重要課題であり、理研とのコデザインの中で10年後も引き続き重要と思われる既存のアプリケーションをベースに、今後新たに必要となる現時点では未知のワークロードも見据え、エネルギー効率の最適化について議論する。

(4) 日本国内パートナや日本固有技術との連携について、可能性を模索する。

＜技術背景＞

昨年度のレポートでも述べたように、計算科学とデータ科学の統合により、スーパーコンピューティングは新たな時代に突入している。これらの利用技術の進化は、富岳NEXTが導入される2028年に向かって想像を超える速度で進化を続け、それを支えるための計算基盤の重要性は益々高まると考える。一方、半導体技術はムーアの法則の限界がより顕在化し、従来技術の延長線上で電力性能比を向上させることは不可能となっている。

NVIDIAは、20年以上にわたりGPUを開発し、理論性能およびアプリケーション実行性能の継続的な向上を達成し、アーキテクチャおよびエコシステムの発展に取り組んできた。最近では、GPU単体の性能向上のみならず、CPU-GPU-DPUを統合し、データセンターレベルでの実行性能を向上させるためのプラットフォーム開発に注力し、時々刻々進化するアプリケーションの多様化に対応している。

富岳NEXTのフィジビリティスタディにおいては、NIVDIAのアーキテクチャロードマップをベースに、理研や日本企業との最先端の技術連携を視野に入れ、グローバルなリーダーシップと市場展開を目指したあらゆる可能性を模索している。

＜2028年に向けてのロードマップ＞

NVIDIAはこれまで、ほぼ2年ごとにアーキテクチャを更新し、前世代の2倍以上の性能効率向上を実現してきた。製品においては、CPU-GPU-DPU-ネットワークのラインアップを持ち、1年ごとにプラットフォームの更新を継続している。

アプリケーションレベルでの実行性能向上には、ハードウェアの進化のみならず、ライブラリやツールなどのソフトウェアの改良や更新が不可欠である。NVIDIAでは、同じハードウェアアーキテクチャ世代の中でも、ソフトウェアの継続的な開発によって、年々実行性能を向上させている。

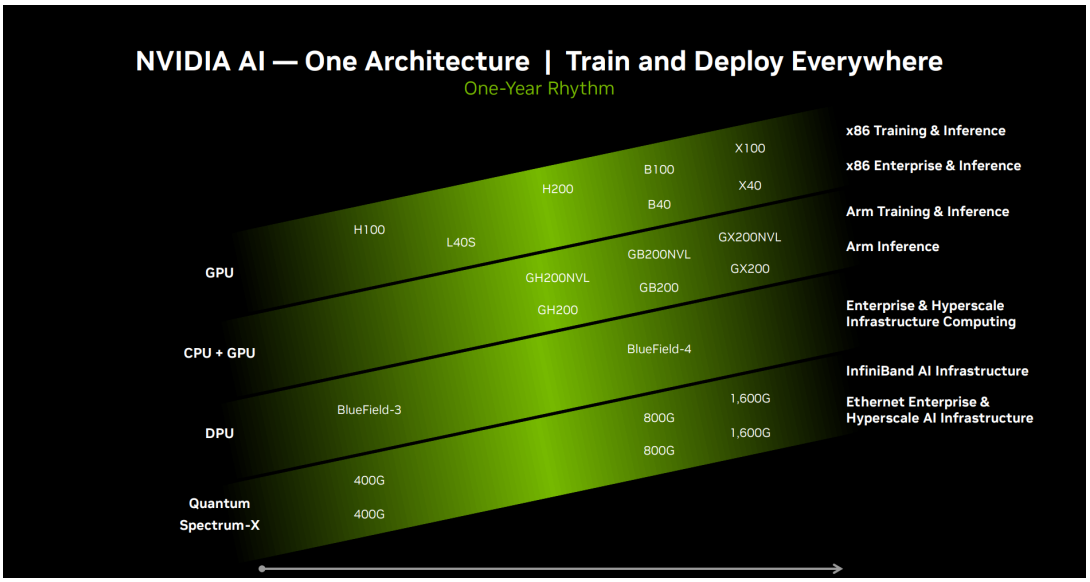


図 2.6.1.1

＜本年度の活動＞

本年度は、2028年頃に想定されるNVIDIAアーキテクテクチャの技術要素の選択肢、推定性能、消費電力等に関する見通しを提示した。ただし期待される性能については、選択される技術要素の革新性度合いにより差異が想定されるため、その旨も含め提示した。個々の選択肢について、グローバルなリーダーシップを目指して日本国内の企業や技術とどのような連携が可能か、本プロジェクトの指針にそって引き続き日本国内の企業と議論を進めていきたい。

2.7 アーキテクチャ調査研究サブグループ 6（HPE）

2.7.1 調査研究の概要

HPEでは、ポストエクサスケールのシステムアーキテクチャに対する複数のアプローチを検討している。

直近10年の後半 (2025年～2030年) には、気候変動などの地球規模の課題やスマートシティを含む社会全体の目標に向けた取り組みにおいて、世界が必要とする科学の問題を究明していくスピードを維持するために、HPCとAI主導のワークロードとワークフローの組み合わせが不可欠である。HPEは、これらの課題に対応するためにはインテリジェントなHPCシステムの活用が極めて重要になると考えており、ここで説明する複数のテクノロジに投資している。そうしたテクノロジはさまざまな組織や企業との連携を深めることによって、影響力を高められると見ている。

HPEは、さまざまなノードアーキテクチャに基づくコア処理技術の開発を、CPU/GPUの主要ベンダであるAMD、Intel、およびNVIDIAと連携して取り組んでいる。これらのベンダは、PCIeやCXLなどの業界標準の接続規格と比較して、ノード内の直接接続されたチップ内で優れたパフォーマンスを発揮する独自のローカルインターコネクト (xGMI、UPI、NVLink) の開発を続けている。HPEでは、次世代計算基盤の開発期間を通してこの取り組みが継続されることを期待している。HPEは、新たな代替プロセッサテクノロジの台頭を引き続き注視しているが、主要ベンダのこれまでの投資規模を考えると、ウェハースケール統合のような画期的なアプローチだけが対抗できると見ており、この開発期間中にウェハースケール統合に向けた開発努力がどこまで実を結ぶか注視している。このアプローチを推し進めれば、個別のプロセッサ間の第1レベルのインターコネクトを根本的に排除し、ウェハ上の各プロセッサを高帯域幅/低レイテンシ/省電力で接続できるようになる。HPE Slingshotイーサネットの機能は、ウェハースケールデバイスに直接統合でき、ウェハーレベルのプロセッサをスケールアウトすることができる。

HPEは、高度なプロセッサノードを結合してバランスのとれたシステムアーキテクチャ構築を実現するために、HPE Slingshotイーサネットのさらなる進化を目指す研究開発に投資している。これには、チップレットレベルのプロセッサとの統合の可能性の検証やノード間通信のためのシリコンフォトニクスの使用が含まれている。HPEは、日本のプロセッサ開発メーカーと同様の方法で協力することに前向きである。HPEは、Slingshotによるエクサスケールの成功から得られた知識を活用して、ハイパフォーマンスなイーサネットの共通オープン標準を確立するために、Ultra Ethernet Consortium - UEC (https://ultraethernet.org/)のメンバとしても活動している。UECでの活動は、プロトコル、フィジカル、リンク、トランスポート、ソフトウェア、ストレージ、コンプライアンスといった各重点分野のワーキンググループに分かれている。UECのウェブサイトには、それぞれの詳細が掲載されており、HPEは、理化学研究所や理化学研究所が提携している他のテクノロジパートナーがUECへの参画を検討するよう強く提案する。UECでの活動は、HPCとAIのワークロードの需要に焦点を当てており、次世代計算基盤に求められるアプリケーション領域と一致すると考えられる。UCEへの参画により、次世代計算基盤プログラムの一環としてテクノロジを使用する際に発生する可能性のあるリスク(非標準・非オープン技術採用によるガラパゴス化など)を軽減することにつながる。また、UECはHPCとAI向けのオープンな標準イーサネットの確立を目指すため、コンピューティングノードの一部として構成されるネットワーク・アダプタとスイッチング・インフラストラクチャ間ネットワーキングのためのコンポーネント／技術を提供する日本のテクノロジと協力できる可能性がある。

HPEは、次の分野でのコラボレーションへの注力が、次世代計算基盤のアーキテクチャにもたらす影響にとって最も生産的であると考えている。

- HPE Slingshot イーサーネットインターコネクト、ならびに現在の開発方針の一例であるシステムオンチップ (SOC) アプローチから、コンピューティングノードあたりの処理速度を最大 50 倍にでき得る完全なウェハースケール統合を活用するための極めて画期的なアプローチにまで至る、処理技術との緊密な統合
- システム管理や HPC と AI 向けの生産性に優れたプログラミング環境を含むシステムソフトウェアスタック

次世代計算基盤の処理ノードについては現在、ローカルメモリ帯域幅に関して20TB/s達成を目標にしており、システムのバランスを保つためには、インターコネクト帯域幅の目標をその5%の範囲 (すなわち 1TB/s) に収めることを目指している。今後5年から10年の間には業界標準のイーサネットが800GB/sに達する見込みであるため、8～10本の密結合したSlingshotイーサーネットチャネルを各処理ノードで有効にすることを見据えている。ネットワークエンドポイントは、参照の局所性を維持するために、処理ノードのオンチップネットワーク上に分散される (現在、NICはマルチGPUノード上に分散されている)。この密度を実現するには、集積シリコンフォトニクスが必要になることが考えられ、処理デバイスやインターコネクトスイッチ用のデバイスと同一パッケージ化することも視野に入れている。

10年後に、どのリンクテクノロジが最大の価値を提供するかについては、まだ確実な見通しを立てることはできない。現在利用可能な最速のリンクテクノロジ (1600Gbps) が、その時点でも最も費用対効果が高いとは限らない。HPEの設計は、ノードごとに複数のエンドポイントとNICごとに複数のリンクを持たせ、高度に並列化されている。これは、各ノードで実行されているアプリケーションの優れた同時実行性を反映しており、多くのプロセスがネットワークトラフィックを生成していることを示している。ネットワーク設計におけるこのような柔軟性を利用して、必要なレベルのグローバルな帯域幅を費用対効果が高く提供するための最適化を図ることができる。

HPEでは、ベースプロセッサおよびインターコネクトテクノロジに加えて、次世代システムのソフトウェア要件を調査している。

HPEは、HPE Performance Cluster Manager(HPCM)とCray System Management(CSM)で従来のHPCシステム管理環境と現在のHPE製品で使用されているクラウド運用環境を組み合わせた新しい環境を構築する予定である。また、日本でのテクノロジパートナーとのコラボレーションの機会を最大限に活かすために、HPCシステム管理のための新しいオープンソースプロジェクト(https://github.com/OpenCHAMI)に参画している。

プログラミング環境に関しては、HPEは従来の大規模なHPCアプリケーションの強化を図るために、HPE Cray Programming Environment (CPE) を進化させる取り組みを継続している。HPEは、AI/MLコードを「大規模に」実行し、多数のノードを使用したモデルトレーニングを推進するために、HPE Machine Learning Development Environment (MLDE) の開発も進めている。MLDEはオープンソースコード (https://github.com/determined-ai/determined)をベースとしているため、特に日本で開発された新しいプロセッサノードアーキテクチャのサポートに関して、コラボレーションの機会が明らかに広がっている。

2.8 アーキテクチャ調査研究サブグループ 7（ARM）

2.8.1 企業紹介と調査研究範囲

Armは、世界の高度なデジタル製品の中心となる、半導体の知的財産（IP）と関連技術を設計している。Armのビジネスモデルの基礎となるのが、パートナーシップアプローチである。ArmはCPUを直接販売せず、その技術と設計を膨大な企業にライセンス供与している。Armは、アーキテクチャ（例：ARMv9）、特定のマイクロアーキテクチャ（例：Neoverse V1）、オンチップインターコネクト技術（例：CMN-700）、モバイルGPU（例：Mali）などのIPをライセンス供与している。最近では、最先端のプロセスノードをターゲットとした、SoCの設計プロセスを簡素化する製品として、パートナが共通で統合するIPを提供するコンピュート・サブシステム（CSS）設計のリリースを開始している。

Armは、最大192個のCPUを統合して動作している、データセンタ/HPC市場向けのIPを提供している。CPU性能のスケーリングには収穫逓減が発生するため、将来的なIPには、コア数の増加と、より緊密に統合されたアクセラレータへの対応が要求される。将来的なデータセンタ/HPCプロセッサ向けのスケーラブル・システムのアーキテクチャ/マイクロアーキテクチャの側面の調査研究に関して、我々は膨大な経験を持っている。エコシステムで独自のポジションを確立している弊社には、アーキテクチャ、マイクロアーキテクチャ、SoCサブシステム、関連ソフトウェアの協調設計において果たせる役割がある。

Armのビジネスの性質上、今回のフィジビリティスタディの調査研究範囲を、Arm IPの評価とフィジビリティスタディのワークロードへの適用に限定した。そのため、フルシステム分析は実施しておらず、コア性能、SoCのインターコネクト、メモリバランスの分析に重点を置いている。将来的には、特定のシステム・インテグレータやアクセラレータ・ベンダと共に分析内容を拡大したいと考えている。

2.8.1.1 基盤技術のトレンド

ムーアの法則が減速する中、効率化とソケットレベル性能の継続的な向上を両立させるため、緊密な統合とより高密度な設計に注目が集まっている。こうした統合のトレンドにより、システムオンチップ（SoC）設計は、単一パッケージ内の複数のチップレットに分割される形で、システム・イン・パッケージ（SiP）設計へと移行している。データセンタ/HPCシステムには、コストを最小限に抑えつつ、限られたパワー・バジェットでより高いパフォーマンスが要求される。ハイパースケーラは、総所有コスト（TCO）あたりのパフォーマンスの最大化を目標としており、より高レベルのCPU統合でコア密度を向上させることで、エンクロージャやネットワーキングといったシステムの「接着剤」を削減しつつ、共有リソースは、最高性能ではなく平均の利用率向上のために分割する。上記にトランジスタ・スケーリングの鈍化が相まって、数百個のCPU、数GBのオンチップSRAMと、高度な2.x/3D技術を使用した数TB/秒のメモリ帯域幅のシステム統合に対する関心が高まっている。

目標は、データセンタ/HPCプロバイダのウェアハウス規模のコンピュータを対象に、全体的なTCOあたりパフォーマンスを向上させるために、単一のプロセッサに可能な限り高い演算能力を統合することである。SiPプロセッサ設計を実現する最近の技術としては、以下が挙げられる。

- 単一のパッケージ基板上に複数のチップを集積する、ファウンドリの提供する高度なパッケージング技術
- チップの境界を越えて広がるシステム・コンポーネント間で通信を行うための、一つのパッケージ内の複数チップ間の共通インターフェイスとしてのマルチチップ規格。マルチチップは、リソース（コア、キャッシュ）の拡大とヘテロジニアス統合によるコスト削減の両方、さらには、演算チップとアクセラレーション・チップをパッケージ内で分離することで、用途特化の負担軽減にも使用される。
- 機械学習などの高性能ワークロードで要求される、上昇した熱設計点の設計を収容するために、データセンタ環境では水冷技術が採用されている。この技術はすでに HPC システムで広く使用されているが、

商用データセンタでの採用により、より先鋭的なパッケージング設計や幅広い演算密度の使用を推し進める方向に熱エンベロープを拡大する可能性がある。

これと同時に、クラウドの弾力的な性質とネットワーキング・レイテンシや帯域幅の進化を受け、利用可能なリソースの全体的な使用率を高めるべく、メモリ、アクセラレータ、ストレージなどのシステムリソースの分割が進んでいる。

これらとは対極的なコンピューティング要求として、各種の社会システムはデータの増加に対応するため、相互接続型のインテリジェンスをエッジ部に導入しつつあり、人々の暮らす都市や工場、農場、環境を計測し、持続可能性、効率性、安全性、生産性を向上させるという、大規模な機会が生まれている。機械学習（ML）分野の飛躍的な進化に基づき、科学者たちは「ソフトウェア定義センサ」を提唱し、インフラストラクチャ全体を通して「エッジコンピューティング」のコードを導入する見通しである。これにより、生データの輸送コストが軽減されると同時に、信頼性、プライバシ、データ主権に関わる懸念も解消される。

ここでエッジコンピューティングとは、デバイスのエンドポイント上、および、こうしたエンドポイントとクラウドを接続するゲートウェイ/ネットワーク要素上でのオペレーションと定義する。多くの場合、こうした形態のエッジコンピューティングは、制約のあるハードウェア上で行われるが、制約のあるハードウェアでも、相互接続とインテリジェント化は進んでおり、複数のタスクを処理できるようになってきている。フィルタリング、分析、集計をデータソースのより近くで実行することで、業界はこの汎用演算機能を活用するモデルに収束しつつある。こうしたデバイスは、マイクロコントローラのようなローエンドシステムから、GoogleのEdge TPUなどの統合型MLアクセラレーションを採用したシステム、NVIDIAのJetsonプラットフォームのようなGPUと結合した組み込みコアまで、幅広い演算能力に対応する。これと同時に、アプリケーションが高度に分散化したヘテロジニアスなインフラストラクチャ上に展開されつつあるため、アプリケーションの管理とオーケストレーションは困難な課題となっている。

クラウドやHPCデータセンタへデータを大量に送り返すのとは対照的に、科学コミュニティの研究者の間では、インテリジェントなセンサやアクチュエータとローカルの組み込み計算リソースを組み合わせ、計算ステアリングと前処理の両方に対応するケースが増加している。HPCセンタがデジタルツインをホストし、デジタルツインはエッジ・インテリジェンスによってキャリブレーションされ、エッジ・インテリジェンスに計算ステアリングを提供するという、新たなワークフローが台頭している。こうした新たなワークフローは、HPCデータセンタ、クラウド、組み込みコンピューティングのどの環境に現存するものとも異なっており、信頼性、セキュリティ、プライバシ、エネルギー効率の問題に対応するため、特別な配慮を必要とする。

我々は、こうした補完的なトレンドには多くの共通なコンポーネントと課題があると考えており、センサからスーパーコンピュータまでを幅広く網羅したArmのエコシステムは、この新しく多様に接続されたコンピューティングのファブリックにコンポーネントを提供しつつ、それらにまたがる統一化を実現する標準規格に貢献できるユニークなポジションにあると確信している。

歴史を見ると、新たなプロセスノード技術は、周波数や消費電力とトランジスタ密度で様々な利点を実現してきた。最近の歴史の示すところでは、こうした利点は7nm未満では大幅に低減しつつあり、コア・マイクロアーキテクチャの技術革新によってパフォーマンスやエネルギー効率は向上する一方で、そのマイクロアーキテクチャが実装される技術ノードによる性能の改善は極めて限定的である。さらに、（プロセスの電圧スケーリングによる収穫逓減により）消費電力はマイナスの影響を受けていると考えられるが、一般的に、新たな技術ノードが実現する演算密度に電力供給と放熱のリソースが追いついていない。

2.8.1.2 Arm の技術ロードマップ

注記：下記のロードマップは、ArmのWebサイトの公開情報に基づいている。より詳細なバージョンや将来的なロードマップ項目は、秘密保持契約に基づき公開される。

2.8.1.2.1 アーキテクチャのロードマップ

Armのアーキテクチャは、過去10年間で著しく進化しており、ハイパフォーマンス・コンピューティングや機械学習のワークロードの実行能力を向上させてきた。こうした取り組みの発端となったのが、Scalable Vector Extension（SVE）の導入である。SVEにより、コードの再コンパイルを必要としない形で、より広範なマイクロアーキテクチャのベクトル長のセットに対し、ベクトル長に依存しないアプローチを提供する道筋が開けた。富士通のA64fxに初めて搭載されたSVEは現在、その第二世代アーキテクチャ（SVE2）が公開中で、データセンタ・アプリケーションを対象としたNeoverseコアの標準的なコンポーネントになっている。

機械学習とハイパフォーマンス・コンピューティング・アプリケーションのさらなる強化を目指し、Armは近年Scalable Matrix Extensions（SME）の研究を行っている。このSMEは、2次元行列タイル、2つのベクトルの外積をタイルに蓄積する命令、タイル幅に一致するベクトル長でSVE2の実行を許可するストリーミング機能、のアーキテクチャ状態を提供する。このアーキテクチャ設計の第二世代（SME2）では、当初のフォーカスであった外積と行列対行列の乗算を越えて、SMEから恩恵を得られるアプリケーションの数を増やすために、そのアーキテクチャ内で実行可能な命令を拡張している。

SVE、SVE2、SME、SME2に関する詳細情報は、ArmのWebサイト（弊所の技術文書内）で公開している。

2.8.1.2.2 データセンタ/HPC 向け Neoverse ロードマップ

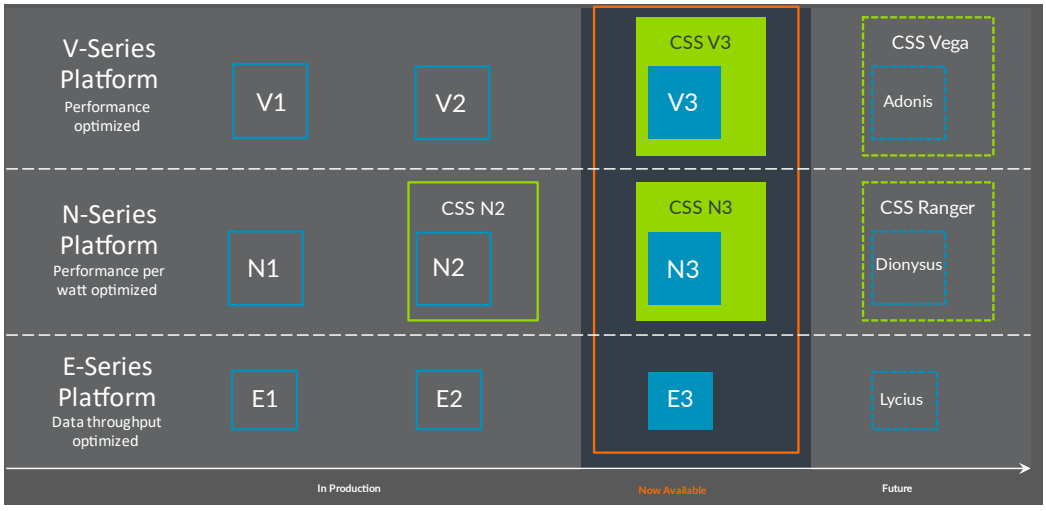


図 2.8.1.1 インフラストラクチャ/HPC 向けのコア/演算サブシステム

図 2.8.1.1の通り、Armは2024年2月、データセンタ/HPCアプリケーションでパフォーマンスに最適化された演算能力を実現することで、ワットあたりパフォーマンスの水準を高める、V3コアおよび関連する演算サブシステムの提供開始を発表した。このコアとサブシステムの設計では、実環境のワークロードでの総所有コストのポテンシャルを最大化できるよう、チップレット構成を念頭に置いて設計されている。

この演算サブシステムの設計は、設計開始からテープアウトまで最短9カ月という、最速の市場投入期間を

実現できることを自ら証明した。あるパートナは、演算サブシステムを使用した結果、推定80人年分のエンジニアリング作業が短縮された。CSS V3の演算サブシステムは、最近発表されたArm Chiplet System Architecture（CSA）に準拠している。CSS V3は、ソケットあたり最大128コアをサポートし、メモリはDDR5、LPDDR5、HBM3の技術に対応するほか、PCIe Gen5とCXL 3.0もサポートする。

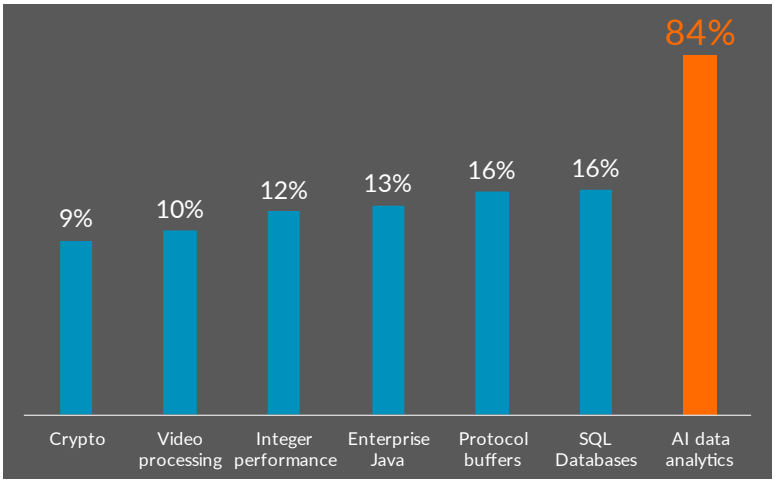


図 2.8.1.2 シミュレーション（log 表示）で測定された、Neoverse V2 に対する Neoverse V3 での向上率（%）

V3設計の初期シミュレーションによると、インフラストラクチャ・ベンチマーク全体でパフォーマンスの向上が示されており、AIデータ分析のワークロードでは大幅な向上が確認された。これは主要パートナ各社との緊密な共同設計によるもので、これによって、基本のマイクロアーキテクチャの変更とキャッシュ階層のバランス調整が行われ、関連するメモリ帯域幅が向上している。

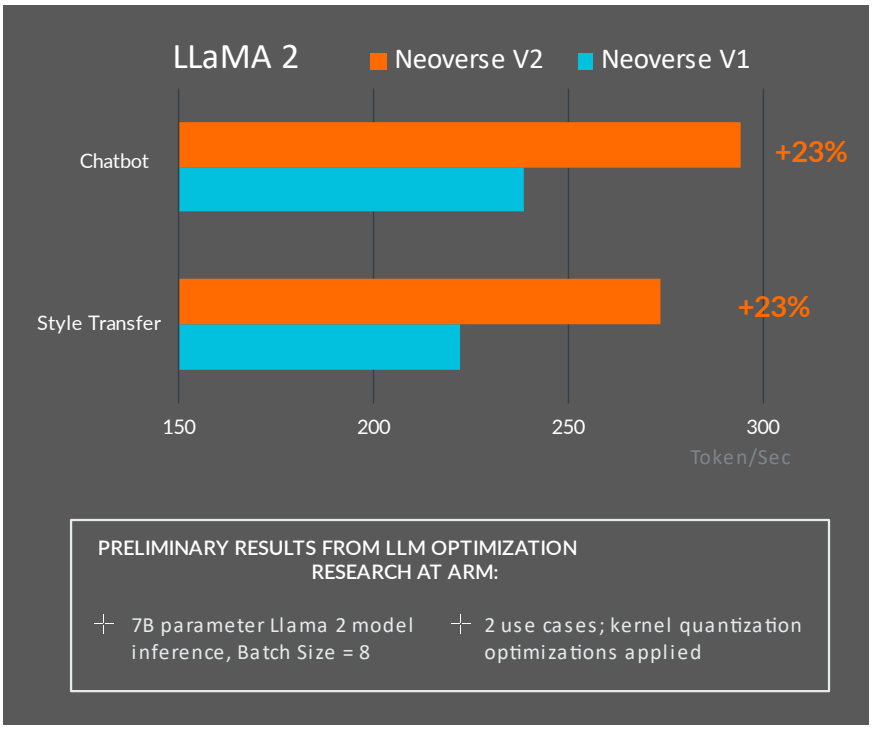


図 2.8.1.3 LLaMA 2 の暫定的な結果

より具体的には、Neoverse V2上でLLaMA 2を用いた我々の分析では、図 2.8.1.3に見られる向上を示した。

2.8.2 経過報告

2023年9月、Armは、以下の目標を念頭に置いて、フィジビリティスタディに着手した。

- フィジビリティスタディの対象ワークロードに対する、現在および将来のアーキテクチャ機能の適用性の分析
- 本アーキテクチャのマイクロアーキテクチャ実装の評価・調整で使用される弊社の標準的な性能リグレッション・スイートに対し、フィジビリティスタディのワークロードを合体
- フィジビリティスタディの対象ワークロードに対する、弊社の内製コンパイラ、ライブラリ、ランタイム動作の適用性を分析

リソースの制約と限られた時間の中で分析を行う必要があったため、今回の調査は6つのHPCアプリケーションのうち4つと、二つのMLワークロードが中心となった。弊社の業務の性質上、単一SoCに特化した分析となっており、マルチノードやマルチチップレットの分析は行っていない。ワークロードの構築にはArmのLinux用コンパイラ、Arm Performance Librariesを使用し、物理ハードウェア（利用可能な場合）とシミュレーション技術を組み合わせて、現在および将来的に想定されるNeoverseコアでワークロードを評価した。そして、パフォーマンスイベントを使用し、アプリケーション全体とアプリケーション内のマーク付きカーネルの両方を対象に、単一コアから64コア構成までのコード実行を分析した。このデータに基づき、演算強度とFLOPs使用率をルーフライン図に記入し、ワークロードが本質的にコンピュートバウンドなのかメモリバウンドなのかを理解することで、代替的なメモリ技術、インターコネクトのトポロジ、将来的なマイクロアーキテクチャ実装により、将来の構成でどのようにスケーリングされるかを理解することが可能となる。

前節で言及した通り、Armは現在、Scalable Matrix Extension（SME）と呼ばれる最新のアーキテクチャ拡張機能を開発中であり、これは主に機械学習ワークロードを対象としているが、他のアプリケーションも向上させる可能性がある。フィジビリティスタディのワークロードに対するSMEアーキテクチャの適用性の評価は、我々が今回の調査に参加した主たる動機の1つである。

Armアーキテクチャの具体的な実装やSMEアーキテクチャに関する詳細情報の多くは現時点では公開されておらず、今回の分析結果は、秘密保持契約に基づいてのみ入手可能であり、本文書には掲載していない。

2.8.2.1 ツール開発

- 異なるメモリ技術とトポロジを持つ将来的な IP を評価するため、gem5 のインターコネクトモデルと内部のマイクロアーキテクチャ・シミュレーションを統合（現在進行中）
- SVE2 と SME を実装するため、DynamoRIO ARMIE を拡張（現在進行中）
- ストリーミング SVE の実行を分析するため、DynamoRIO を拡張（現在進行中）
- SME の組み込み関数を使用できるよう、ACfL Fortran を拡張（完了）
- 扱いやすい実行時間でより大規模なワークロードに対応できるよう、Arm の社内シミュレーション・ツール内でチェックポイント/リスタートを強化（今後の課題）

2.8.2.2 ワークロードの高レベル分析

要約すると、SALMON、SCALE-LETKF、LQCD-HMC-DWF向けのHPCアプリケーション・カーネルはいず

れも、FP64表現を厳格に使用していると考えられる。GENESISは、FP32モードで実行されていると思われる。従って、SMEの外積からメリットを得られる可能性は、存在したとしても低いと思われる。SME2でのストリーミングSVEは、より広範なオペレーションのセットをサポートすることで、より有益なものと期待される。

これとは対照的に、機械学習ランタイム用のSME2固有の最適化に関しては、積極的な取り組みが行われていて、生成AIモデルを含むMLモデルの場合、これにフィードするシステムが適切に構成されていると仮定するなら、大幅な向上が実現できるという証拠が早期から存在する。

Arm Performance Librariesは、SALMON（ランタイムの18%）とSCALE-LETKF（13%）の両方で使用されており、ここでは少なくともある程度のストリーミングSVE2を含められる余地が存在する。SME（ストリーミングSVE2を含む）の残りの最適化作業は、コンパイラによって行われる必要がある。これは、SVEの採用に関してHPCアプリケーションで起こったことの傾向と概ね一致している。すなわち、大半は、ベクトル処理のアーキテクチャ固有のチューニング作業をコンパイラに依存していて、アーキテクチャのトレンド要件に対応するための極めて明確かつ正当な理由が存在しない限り、アップデートされたソースコードの開発を回避している。しかし、広く使われているアプリケーションの中には、ソースコードを特定のアーキテクチャやベクトル実装に最適化しているものが必ずいくつかあり、そのようなものが最も恩恵を受ける可能性がある。

一般的に、アプリケーションのスケーリングに対するアーキテクチャ上の障害は確認されておらず、大半のワークロードは、明らかにメモリバインドに偏っていた。そのため、バランスの取れた設計を実現するためのトレードオフは、メモリ階層のスケーリングのコスト対コアの数と能力で決まる。ここでの課題の一部は、三次元実装によって目標期間内に緩和される可能性があるが、その際には、電力供給と冷却技術を向上させる必要がある。

SVE2、SME、ストリーミングSVE2を使ったArm Performance Libraries（HPCアプリケーション用）とArm Compute Libraries（機械学習用）の最適化作業は、現在も積極的に進められている。特に、機械学習におけるより広範な演算子のサポートや、より具体的には最近のアーキテクチャの改良をサポートするために、さらに多くの作業を行えるという証拠がいくつかある。同様に、SVE2向けのコンパイラ開発は現在も進行中であり、コンパイラ内のSMEに関する作業（ストリーミングSVE2のサポートなど）は始まったばかりで、組み込み関数のサポートに限定されており、コード生成は行われていない。今後、こうした機能が利用可能になるまでの期間、開発を継続することで、HPCのフィジビリティスタディのワークロードを含む、より広範なアプリケーションへの適用性が向上し、理想としてはアプリケーションの演算強度が向上すると期待される。

2.8.3 今後の課題

2.8.3.1 精緻な分析

今回の調査は時間が限られており、弊社の内部リソースの制約もあったため、ハイレベル分析については、対象ワークロードの一部にしか実施できなかった。特に障害となったのは、この調査の成果としてFORTRANコンパイラのサポート作業が進行中であるにも関わらず、比較的新しいアーキテクチャ機能（SMEなど）に対するFORTRANコンパイラのサポートだった。パフォーマンスと効率性を最大化する上で、どのような新しいアーキテクチャやマイクロアーキテクチャのパラメータを協調設計できるかを理解するには、アプリケーションにもっと時間をかけ、具体的なパフォーマンスのボトルネックを掘り下げる必要がある。我々は来年にかけて、こうしたより深い調査を計画している。

2.8.3.2 SME/SSVE 分析用の動的バイナリ変換ツール

今回のプロジェクトの一環として、SVE2とSME（MOPAとSSVE）のオペレーションをエミュレートするため、我々はDynamoRIOのArm Instruction Emulator（ArmIE）プラグインの拡張に着手した。初期フェーズでは、この作業を完了できなかったが、作業は継続しており、エミュレーションの上に分析ツールを構築することで、

ストリーミングSVE2のオペレーションがパフォーマンス上のメリットをもたらすであろう領域を特定していく計画である。この情報はその後、手作業による最適化や、弊社のコンパイラ/ライブラリチームにフィードバックできる。

2.8.3.3　スケーラブルなシミュレーション

この研究の大きな制約となったのは、我々のシミュレーション環境でフルスケールのフルアプリケーションの実行を様々なメモリ技術とメッシュトポロジで実行することに対する我々の能力だった。我々は詳細シミュレーションのチェックポイント/リスタートを大幅に効率化する技術に取り組み始めていて、メッシュ上のフルSoCの干渉パターンを分析できるよう、シミュレーションのスケーリング手法についても考え始めている。さらに、今回の取り組みを拡大し、サードパーティのシミュレーション・データを採用することで、緊密に結合したアクセラレータ（DPU、NPU、GPUなど）が、チップレット技術を用いて同一SoC上にパッケージされた際の相互動作を理解したいと考えている。

2.8.3.4　生成 AI のメモリ干渉パターン

生成AIモデルは、主にメモリ階層がボトルネックとなっていることが判明している。システムアーキテクチャの協調設計に関する理解を深めて、このボトルネックをより軽減するためには、システムアーキテクチャをどのように協調設計すればよいかをよりよく理解するために、いくつかの異なるフレームワークを使って、この問題をより細粒度で詳細に調査する予定である。

2.8.3.5　アクセラレータ・インターフェイス

CPUとアクセラレータのより緊密な連携に関する調査、特に生成AIモデル向けの演算子の分配を見ていく。特定のワークロードの早期分析では、CPUとアクセラレータの間で演算のバランスを取ることで、最適な効率とパフォーマンスを実現できることが判明しており、与えられたワークロードを使用し、この分析をより深く掘り下げたいと考えている。

2.9 まとめと今後の課題

アーキテクチャ調査研究グループでは、グループ構成員およびA0～A7のサブグループにより、技術動向調査やアーキテクチャ検討指針の検討に加え、ベンダサブグループの提案する複数のアーキテクチャ候補の詳細化と比較・評価を中心に、以下を実施した。

- サブグループ A0：半導体技術・パッケージング技術・アーキテクチャの技術動向の調査、ベンチマークプログラムの特徴解析によるワークロード分析、次世代計算基盤への要件や評価指標の整理、システムとしてあるべき姿の検討、他のサブグループ調査研究の方向付けと推進、内容のレビューやフィードバック。アーキテクチャ候補の絞り込み
- サブグループ A1～A7：提案アーキテクチャ候補の詳細化と評価。特にノードアーキテクチャの詳細化とその性能・コスト等の見積。理研の選定したベンチマークによる性能推定

今後の課題は、以下の通りである。

- アーキテクチャ候補のさらなる絞り込みとベンチマーク性能推定
- 開発を想定するシステム構成や規模などの検討
- 開発ロードマップの検討
- システムソフトウェア・ライブラリ調査研究グループおよびアプリケーション調査研究グループと連携した、アーキテクチャ評価