将来の HPC アーキテクチャに関する私見

牧野淳一郎 †

「今後の HPC 技術の研究開発を検討する作業部会」アプリケーション作業部会のスペック検討サブ WG での経験をふまえて、今後の HPC アーキテクチャの方向についての私見を述べる。

Personal view on the future HPC architecture

Junichiro MAKINO †

I discuss the direction of the reserch and development of HPC architecture of the future, based on the experience of the specification sub-WG, within the application WG, under the WG for the future HPC reserch and development under MEXT.

1. はじめに

昨年 7 月からの「今後の HPC 技術の研究開発を検討する作業部会」に、アプリケーション作業部会のスペック検討サブ WG のまとめ役の一人として関わることになった。その経験も踏まえて、今後の HPC アーキテクチャに関する個人的な意見を述べたい。

2. 電力効率とスケーラビリティ

現在の汎用プロセッサの延長の方向で HPC 向けの並列度の高いシステムを構築することは困難ないし非現実的になってきている。この理由は 2 つある。

一つは電力効率の問題である。Top500 の 1 位にくるシステムの消費電力は、この 10 年間で 10 倍近く増加した。半導体技術の進歩、アーキテクチャの変化が同様なものなら、今後 10 年でさらに 10 倍程度増大し、100MW 近い消費電力となる。

これは少なくとも日本では電気代のほうが計算機調達コストよりも大きくなることを意味する。なお、GPGPU 等の比較的ハードウェアが安価なシステムの場合、既に電力コストのほうがハードウェアコストよりも大きくなっている。

つまり、半導体技術の向上による以上に電力当り性能を向上させることができないと、計算機の性能の進歩は今後大きくスローダウンすることになる。

もうひとつは、大規模システムが汎用性を失ってきていることである。「京」においても明らかであるように、100 万コア近い大規模並列システムである程度の実行効率が得られるアプリケーションは、非常に並列度が高く、通信粒度が大きなものに限定されている。さらに 100 倍、1000 倍といった性能のシステムを考える時、並列度がさらに 100, 1000 倍、ないしはそれ以上必要だと、実用的な性能がでるアプリケーションがなくなってしまうかもしれない。例えば 3 次元流体計算で、1 次元方向のメッシュ数を N とすると必要なメモリ量は N の 3 乗であるのに対して計算量は少なくとも N の 4 乗である。1 コア当りの計算性能が同じなら、同じ時間で計算を終えるためには 1 コア当りのメッシュ数は 1/N に比例して小さくなる必要があり、これは通信のバンド幅やレイテンシへの要求が厳しくなることを意味する。クロック速度があまり増加せず、チップ内のコア数が増加し、メモリ階層が深くなる結果主記憶のレイテンシが増大し、バンド幅も相対的に小さくなるとすると、隣接コア間通信でもバン

† 東京工業大学大学院理工学研究科理学研究流動機構
Interactive Research Center of Science
Graduate School of Science and Engineering
Tokyo Institute of Technology

ド幅、レイテンシを現在の程度に保つことがそもそも困難である。さらに、同期や縮約等の大域的な通信については、コア数が大きく増えるために隣接コア間に比べてさらに大きくなる。このため、3 次元流体計算のような、現在のところなんとかスケーラビリティを維持出来ているようなアプリケーションでも、コア数がさらに 100, 1000 倍となると実用的な問題サイズではスケールしなくなってくる。

なお、多くの場合にはネットワークの性能による制約よりも主記憶バンド幅の制約が先に現れる。古典的なベクトルプロセッサでは B/F (byte per flops) が 4 であったが、最近のマイクロプロセッサでは 0.25 前後、「京」で 0.5 程度と低下しており、2018 年頃には 0.1 を維持することも困難と予想されている。このため、3 次元流体計算のようなアプリケーションでは予測される実行効率があまり高くならず、結果的にネットワーク性能の問題が表面化してきていない。

3. 外部メモリをもたないアーキテクチャ

アプリケーション作業部会では、多様な分野からの数十のアプリケーションについて、計算機アーキテクチャに要求する特性についての予備的な調査をおこなった。やはり、3 次元流体計算や、有限要素法による構造解析では性能が B/F によって決まるようになっていて、高い実行効率を得るのは困難であった。また、現在知られている計算方法では大域的な FFT を必要とするようなアプリケーションは性能が完全にネットワークバンド幅で決まる。このため高い実効性能を実現するためには非常に高バンド幅のネットワークが必要になり、実装技術的にも消費電力的にも困難となる。

一方、上で述べた、多くのアプリケーションで計算速度に対してそれで実行可能な計算サイズがそれほど大きくならない、という問題は、逆にいうと計算速度の上昇に対して必要なメモリ量は比例しては増えなくなってきているということである。上の 3 次元流体計算でも計算速度に対して必要なメモリ量は 3/4 乗である。もっと弱い、あるいはそもそも計算速度が上がればそれに比例した長時間計算をしたい、つまり必要メモリ量は一定に近いアプリケーションも数多くある。

このようなアプリケーションに対しては、オンチップで搭載可能なメモリにデータを載せることで、ノードレベルでのバンド幅の問題を回避し、さらにアクセスレイテンシも短くする、という解がありえる。これは、連続系の計算ではあるが比較的格子サイズが小さい格子 QCD 計算向けの計算機では QCDOC あたりから採用され、また分子動力学専用計算機 ANTON でも採用されている方式である。 2018 年頃を考えると、単純なメモリであればチップ当り 256MB 程度を搭載することはそれほど困難ではなく、10 万チップでシステムを構成するなら 20TB ものメモリをもつことになる。自由度が 10^{10} から 10^{11} 程度の系ならば扱えるわけである。このような、基本的にはオンチップメモリにデータをもつシステムでも、その内部構成が単純に現在のプロセッサの延長から外部メモリをとっただけであってはトランジスタ効率も電力効率も低く通信レイテンシも大きい。これは、現在のプロセッサが外部メモリに対する実効的なバンド幅を高くする、そのためにはチップ内でかなり無駄なオペレーションをすることもいとわない、という設計思想で構成されているためである。

この典型的な例が多階層キャッシュである。多階層キャッシュは行列乗算のような複数のレベルでブロッキングが可能なアルゴリズムでは有効だが、殆どの HPC アプリケーションは基本方程式や差分化の方法を決めたところでブロッキングの自然な方式が決まり、行列乗算のような任意性があることは少ない。このため、マルチレベルのキャッシュはほとんどの場合無駄にデータを移動して電力消費を増やす効果しかない。また、キャッシュラインベースのアクセスでないと無駄が発生することから、多くの数値計算で必要なストライドアクセスやランダムアクセスで著しい性能低下を引き起こしている。ノード間通信についても、多階層メモリはレイテンシを著しく増大させる。

つまり、オンチップメモリを前提にプロセッサを設計するなら、現在のような多階層キャッシュとは根本的に異なるメモリアーキテクチャを考える必要がある。それにより、初めてそのメリットを生かした高効率なプロセッサが実現できる。

4. おわりに

将来の HPC アーキテクチャを考えるにあたっては、多くのアプリケーションの特性とマッチしたアーキテクチャは何か、また単一のアーキテクチャでなにもかもカバー出来るのか、といった検討が重要である。上で述べた外部メモリをもたないシステムはその一例であり、他にも多様なアーキテクチャの検討が重要である。