



# OpenMP\* 20 周年

並列プログラミングを C/C++ および Fortran プログラマーが利用しやすいものに

Bronis R. de Supinski Livermore Computing CTO

私は、20 年という歳月を経て OpenMP\* がようやく成人したと感じています。OpenMP\* 20 周年のさまざまなイベントは、OpenMP\* 仕様の原点と発展を改めて見直す機会を与えてくれました。OpenMP\* 言語委員会の委員長として (OpenMP\* 3.0 のリリース直後から約 9 年間務めています)、この回顧録ならびに OpenMP\* 組織の現況と将来の計画を発表する機会が得られたことを嬉しく思っています。私は、OpenMP\* がそのルーツに忠実なまま、今後 20 年間さらなる発展を遂げることに、皆さんも同意されることを期待しています。

OpenMP\* は、ループベースの共有メモリー並列処理に対応するため、多くのコンパイラー・メーカーがまだ標準化されていないディレクティブをサポートしていた時代に生まれました。これらのディレクティブは効果的でしたが、重要な要件である移植性に欠けていました。これらの実装固有のディレクティブは、構文 (スペリング) が異なっているだけでなく、しばしばセマンティクスにも微妙な違いがありました。この欠点に気付いていたローレンス・リバモア国立研究所の Mary Zosel 氏は、各メーカーと密接に協議して、共通の構文とセマンティクスについて合意を取り付け、OpenMP\* 1.0 の提供にこぎつけました。さらに、彼らは OpenMP\* の権利を所有し、OpenMP\* 仕様を管理する組織を創立しました。

多くの方は、OpenMP\* のルーツを、単純に最初の仕様で標準化されたループベースの共有メモリー・ディレクティブとしてとらえているでしょう。私は、OpenMP\* のルーツを、コンパイラー・メーカーとシステムができるだけ移植性の高いパフォーマンスを提供することを求めるユーザー組織による、移植性のあるディレクティブ・ベースの並列化と最適化 (プロセス内に制限された) をサポートする取り組みであると考えています。OpenMP\* は、コンパイラーが (スタティック) 解析により引き出すことが困難または不可能であると判断した、アプリケーションに対するパフォーマンスを明示的に表現するプログラマー向けのメカニズムである、と私は思っています。組織的には、OpenMP\* は、すべての主要コンパイラー・メーカー並びに活発で成長を続けるユーザー組織を含む多くの会員を擁し、活気に満ちた活動を行っています。

技術的には、OpenMP\* 仕様は、利用可能なループベースの並列処理ディレクティブを統一するという本来の目標を達成しました。現在も、その目的のために、単純で、適切に定義された構文の改良を続けています。さらに、OpenMP\* は、ループを結合する機能や、並列化されたコードを実行するスレッドの配置を制御する機能など、ディレクティブの改良も続けています。これらの構文は、共有メモリー・アーキテクチャーおよび **コンパイラー** 全体にわたって一貫した強力なパフォーマンスを達成し、OpenMP\* 誕生の動機となった移植性を提供します。

OpenMP\* の発展とともに、いくつかの新たな並列処理も採用されました。私は、コミュニティーに参加する人々がタスクベースの並列処理のメカニズムの標準化が必要であるという話をしているのを聞くとうんざりします。OpenMP\* は 9 年も前から標準化されたメカニズムを提供しています。2008 年に、完全なタスクベースのモデルを採用した OpenMP\* 3.0 仕様がリリースされました。OpenMP\* のタスク実装はまだ改善できると私は考えていますが、現在はどちらとも決めかねる問題に直面しています。一貫して強力なパフォーマンスを提供するモデルが実装されれば OpenMP\* タスクを使用する、とユーザーが述べているのをよく聞きます。また、モデルの採用が進めばタスクの実装を最適化する、とメーカーが述べているのもよく聞きます。別の視点から見た OpenMP\* タスクモデルの問題は、1 つのアプリケーションですべて使用できるように、一連の並列化手法に一貫した構文とセマンティクスを提供するという、OpenMP\* の長所によるものです。このような一般化のサポートには、オーバーヘッドがつきものです。しかし、モデルを改良することで、オーバーヘッドを多少緩和できます。例えば、OpenMP\* 3.1 では、複雑なコンパイラー解析や複雑なプログラム構造に依存しない、最小タスク粒度を指定する **final** 節 (および関連する概念) が追加されています。我々は、将来の OpenMP\* 仕様に取り組むとともに、OpenMP\* タスクのオーバーヘッドを軽減する方法を見い出そうとしています。

OpenMP\* 4.0 リリースでは、別の観点から並列処理をサポートしました。現在の OpenMP\* には、デバイス構文によるアクセラレーター・ベースの並列化と SIMD (またはベクトル) 並列処理のサポートが含まれています。SIMD のサポートは、OpenMP\* の原点を思い起こさせるものです。実装固有のディレクティブによる SIMD 並列処理のサポートは広く普及しており、**ivdep** 形式が最もよく採用されていました。しかし、サポートされている節はさまざまで、一部のコンパイラーではディレクティブの記述方法も異なっていました。さらに、セマンティクスもわずかに異なることが多く、ディレクティブが 1 つのコンパイラーで正しく、別のコンパイラーで正しくないことがありました。20 年前に OpenMP\* の誕生によりループベースのディレクティブの問題が解決したように、SIMD 構文が OpenMP\* で採用されたことで、これらの問題のほとんどは解決しました。もちろん、SIMD ディレクティブの最初の実装は完璧ではなく、我々は改良を続けています。例えば、OpenMP\* 4.5 では、ユーザーが SIMD ベクトル長を指定できる **simdlen** 節が追加されました。

OpenMP\* の **target** 構文は、構造化ブロックの計算をデバイスにオフロードするように指定します。OpenMP\* 4.0 仕様で追加されたこの構文は、GPU のようなデバイス上での効率的な並列処理をサポートします。重要なことは、すべての OpenMP\* 構文は target 領域内で使用できることです。つまり、ディレクティブベースの並列化における数々の進歩をさまざまなデバイスで活用することができます。特定の種類のデバイスと並列化するアルゴリズムに対して最適な並列処理手法を考える必要はありますが、OpenMP\* 構文の直交性は、複数のアーキテクチャーをターゲットにするプログラムの移植性を大いに高めるでしょう。

我々は、現在、2018 年 11 月に公開予定の、OpenMP\* 5.0 仕様に取り組んでいます。すでにいくつかの大幅な拡張を採用しました。OpenMP\* TR4 は、2016 年 11 月以降の OpenMP\* 5.0 における変更点を文書化したものです。我々は、これまでの進歩を文書化した新しい TR5 を 11 月に公開する予定です。TR4 の最も重要な追加機能は、OpenMP\* をツール・インターフェイスで拡張する OMPT です。私は、OpenMP\* 5.0 では、OpenMP\* アプリケーションのデバッグを促進するツール・インターフェイス、OMPD が追加されると予想しています。TR4 では、多くの拡張が採用されましたが、その中でも注目すべきはタスク・リダクションの追加でしょう。

OpenMP\* 5.0 では、TR4 に含まれていなかったいくつかの重要な拡張を追加する予定です。注目すべき点として、OpenMP\* は複雑なメモリー階層のサポートを大幅に強化します。まず、私は、このメモリー割り当てメカニズムが TR5 に含まれることを期待しています。このメカニズムは、複数の種類のメモリー (例えば、DDR4 と HBM) を含むシステムで、特定の変数や動的割り当てにどのメモリーを使用すべきかを指定する直感的なインターフェイスをプログラマーに提供します。さらに、OpenMP\* が複雑なデータ構造のディープコピーをサポートする、強力なシリアル化 / 逆シリアル化インターフェイスを提供することを期待しています。このインターフェイスは、デバイス構文によるデータ依存レイアウトの変換と関連する最適化を可能にします。

前述のとおり、OpenMP\* 組織は健全です。我々は、過去数年間、新しいメンバーとしてコンパイラー・メーカーとユーザー組織の両方を迎えてきました。現在、さらに参加の門戸を広げるために、OpenMP\* 規約の変更について議論しています。OpenMP\* の将来の構想に興味を持たれている多くの方々の参加をお待ちしています。OpenMP\* の詳細は、[openmp.org](http://openmp.org) (英語) を参照してください。

Bronis R. de Supinski 氏は、ローレンス・リバモア国立研究所のスーパーコンピューティング・センター、Livermore Computing の CTO であり、OpenMP\* 言語委員会の委員長でもあります。

この記事は、米国政府機関の後援を受けた研究の説明として準備されたものです。米国政府、ローレンス・リバモア国立研究所、およびこれらの機関の職員のいずれも、本記事で示された情報、機器、製品、プロセスの正確性、完全性、有用性について、あるいはそれらの使用が私有の権利を侵害しないことについて、明示的であれ黙示的であれ、いかなる保証も行わず、いかなる法的責任も負いません。本記事の商標名、商標、メーカーによる特定の商用製品、プロセス、サービスに対する参照は、必ずしも米国政府あるいはローレンス・リバモア国立研究所による支持、推薦、賛同を意味するものではありません。本記事で示された著者の意見および見解は、必ずしも米国政府あるいはローレンス・リバモア国立研究所の意見および見解を述べたり反映したものではなく、広告や製品を推奨する目的に使用されないものとします。この研究、LLNL-MI-732158 は、契約 DE-AC52-07NA27344 の下で、ローレンス・リバモア国立研究所により、米国エネルギー省の後援の下に行われました。