

## 数値シミュレーションにおけるソフトウェア研究開発の動向 — 並列分散型のハードウェアとソフトウェア自動チューニング —

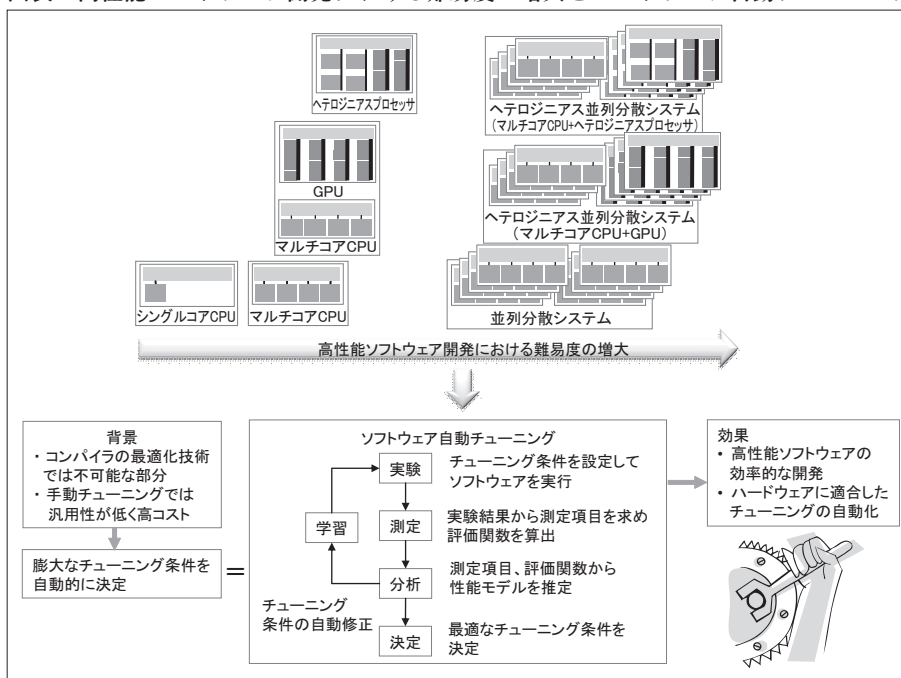
コンピュータの性能向上は、CPU（中央演算処理装置）の動作周波数の向上に依存してきたが、発熱・消費電力やリーク電流増大の問題で頭打ち傾向になった。そのため、近年では、マルチコアプロセッサを用いた並列処理によるコンピュータの性能向上が中心課題とされている。

並列化されたハードウェアの性能を引き出すためには、ハードウェアアーキテクチャに合わせた計算アルゴリズムを選択し、実効効率が向上するようにプログラムを調整しなければならない。この作業はソフトウェアチューニングと呼ばれ、高性能なソフトウェア開発では不可欠な要素である。しかし、ハードウェアアーキテクチャの複雑化により、実効効率の高い数値シミュレーションソフトウェア開発は格段に難しくなっている。

そうした背景から、ソフトウェアの自動チューニング技術を基盤技術として実用化する提案がなされている。ソフトウェア自動チューニングとはハードウェアに合わせたソフトウェアの調整を自動的に行い、実効効率を向上させることである。現在は、行列演算などの数値計算ライブラリを中心にソフトウェア自動チューニングの研究と応用が進められているが、一般の数値シミュレーションで応用するには未だ研究課題は多い。

日本国内では米国と比較しても遜色のない研究が行われていると思われるが、研究から最終的な数値シミュレーションへの応用、実用化までのロードマップ作成や研究者間での役割分担やリソース共有を十分に行っていない。特に今後、日本国内の研究組織には、基盤技術研究と応用研究を効率的に行うための研究マネージメントの再考を期待したい。

図表 高性能ソフトウェア開発における難易度の増大とソフトウェア自動チューニング



参考文献 9、20、26、30～33) を基に科学技術動向研究センターにて作成

# 数値シミュレーションにおける ソフトウェア研究開発の動向

—並列分散型のハードウェアとソフトウェア自動チューニング—

古川 貴雄  
推進分野ユニット

野村 稔  
客員研究官

## 1 はじめに

コンピュータの性能向上は半導体プロセスの微細化によるCPU(中央演算処理装置)の動作周波数に依存してきたが、2005年頃から発熱・消費電力やリーク電流増大の問題で頭打ち傾向になった。そのため、近年では、動作周波数を抑え、汎用処理を実行するコアを複数もつマルチコアプロセッサを用いた並列処理によるコンピュータの性能向上が中心課題とされている<sup>8)</sup>。2000年以降、マルチコアCPUや高速数値演算の可能なGPU(Graphics Processing Unit)、および、汎用処理に対応するコアと数値演算に特化したコアを組み合わせたヘテロジニアスプロセッサが市販されている。こうした並列化によるハードウェアの高性能化が実現される一方、皮肉なことに、実効効率の高い数値シミュレ

ーションソフトウェア開発は格段に難しくなった。

並列化されたハードウェアの性能を引き出すためには、ハードウェアアーキテクチャに合わせた計算アルゴリズムを選択し、実効効率が向上するようにプログラムを調整しなければならない。この作業はソフトウェアチューニングと呼ばれ、高性能なソフトウェア開発では不可欠な要素である。しかし、最近のハードウェアアーキテクチャがあまりにも複雑になり、人手によってソフトウェアをチューニングすることは作業負荷の問題だけでなく技術的にも困難になりつつある。また、開発したソフトウェア資産を維持していくためには、ハードウェアアーキテクチャの世代交代が早いので、その都度、アーキテクチャの変更に合わせて

ソフトウェアの修正とチューニングを行わなければならない。そのために、維持コストが高くなるという問題もある。このようにソフトウェアをハードウェアアーキテクチャの変化に合わせて修正する作業は、数値シミュレーション研究開発を効率的に進める上での大きな障害である。

本稿ではまず、ハードウェアアーキテクチャとソフトウェアアプリケーションの動向を述べ、次に高性能数値シミュレーションのための並列分散型のソフトウェア構成を概観する。そして、今後さらに高性能ソフトウェア開発で重要な役割を果たすであろう自動チューニングについて詳述する。最後に、ソフトウェア基盤技術の研究推進体制の新しい動きについても述べる。

## 2 ハードウェアアーキテクチャとソフトウェアアプリケーションの動向

### 2-1

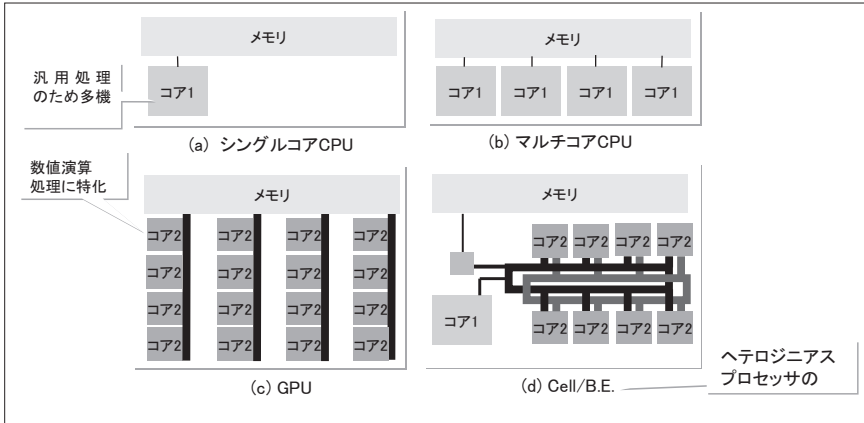
#### 多様化するコモディティプロセッサとその変遷

コモディティプロセッサとは、パーソナルコンピュータやサーバー、ゲーム機用に大量生産される低価格なプロセッサである(マイクロプロセッサ、MPUともいう)。

近年では、科学技術計算用の高性能コンピュータでもコモディティプロセッサを用いた並列分散システムの採用が増えてきている。

図表1(a)～(d)に代表的なコ

図表1 コモディティプロセッサのアーキテクチャ



参考文献 9、20、26) を基に科学技術動向研究センターで作成

コモディティプロセッサのアーキテクチャを示す。(a)シングルコアCPU、(b)マルチコアCPUは汎用処理用に設計され、命令数も多く複雑な処理が可能な処理装置コア1とデータを格納するメモリから構成される。例えば、積の計算といった1つの命令を連続に配置されたデータに適用して演算を高速化する機能や、依存関係のない処理を独立に実行して処理全体を高速化する機能をもつ。

下段の(c) GPUや(d) Cell/B.E. (Broadband Engine)<sup>TM</sup>では特定の数値演算に特化した処理装置であるコア2を中心にした構成が採られている。当初、(c) GPUや(d) Cell/B.E.は3次元コンピュータグラフィックスで多用されるベクトルの内積・外積計算や映像・音声データ圧縮符号化・復号化処理の数値演算を独立に実行して、CPUに負荷を与えないことを目的に開発されていた。しかし、近年では、数値演算能力の高さが注目され、(c) GPUや(d) Cell/B.E.をコアとして数値シミュレーションを行う例が増えている<sup>注1)</sup>。

ヘテロジニアスプロセッサの一つである図表1 (d)のCell/B.E.<sup>注2)</sup>

は、通常のCPUと比較して、以下の改良がなされている<sup>9、28)</sup>。

●プロセッサ内データ転送効率の改善

処理速度向上の点で、メモリ上のデータ読み込み・書き込みとコア間のデータ転送速度がボトルネックとなる。そこで、高速データ転送を実現するために、数値演算の速いコア2間の接続に二重の高速リング状ネットワークを用いている。

●汎用処理用コアと数値演算用コアを用いた処理の効率化と省電力化

処理内容に応じて汎用処理を行うコア1と数値演算を担当する複数のコア2に処理を分けることで実効効率を向上させ、かつ、消費電力を低減している。また、CPUのトランジスタ数の増加は、素子の消費電力増加だけでなくチップの冷却に使用される電力も増やすことになるため、複雑な汎用処理に対応したコア1を増やすのではなく、数値演算を行うコア2を増やすことでチップ全体のトランジスタ数を抑えている。

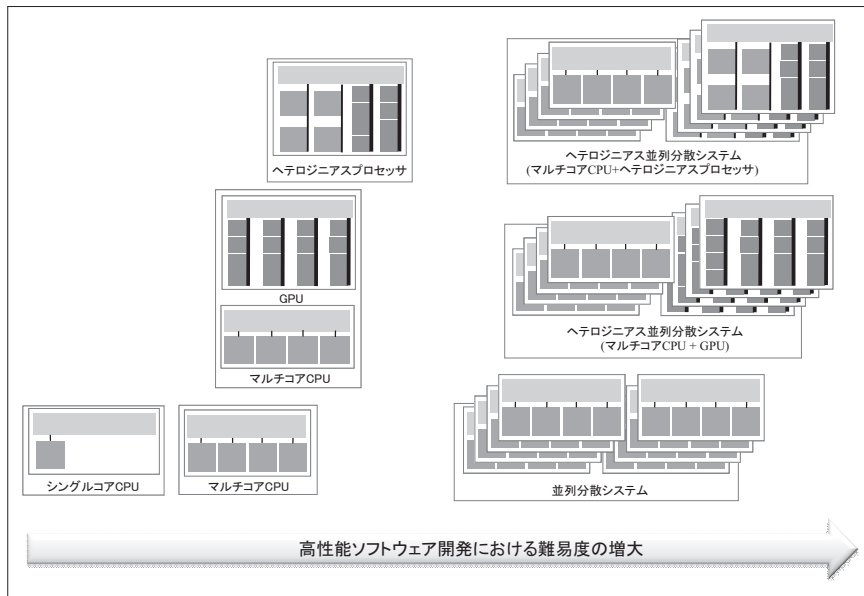
図表2にハードウェアアーキテクチャの変化と高性能ソフトウェア開発の関係を示す。コンピュータの心臓部である演算処理をつかさどるCPUは、過去長期にわたって、シングルコアCPUが全盛であった。その後、マルチコアCPUの登場、そして、GPUとマルチコアCPUをともに使用するなどさまざまなアーキテクチャが採用されてきた。さらに機能の異なるコアが1つのチップ上で構成されるヘテロジニアスプロセッサが現れた。そして、これらのアーキテクチャを混在させて並列化するヘテロジニアス並列分散システムへと変化しつつある。現状は、以上述べたさまざまなハードウェアアーキテクチャが混在されて使用されている。そしてこの混在傾向は継続するといえる。

ヘテロジニアスプロセッサは、IBM社のCell/B.E.<sup>TM</sup>だけでなくAMD社のStreamComputing<sup>21)</sup>、Intel社のLarrabee<sup>23)</sup>の動向から、今後さらに広く普及することが予想される。しかし、現在のアーキテクチャではCPU-GPU間のデータ転送がボトルネックになりがちである。必ずしもすべてのアプリケーション領域で言えるわけではないが、短期的にはCPUと数値計算の高速なGPU、もしくは、ヘテロジニアスプロセッサを混在させたヘテロジニアス並列分散システムによる数値シミュレーションが、同一CPUから構成されるホモジニアス並列分散システムを用いた場合より費用対効果の点でも優れているとも言われている。

これらのソフトウェア開発面での難易度について述べれば、シングルコアCPUと比較して、ヘテロジニアス並列分散システムの高性能

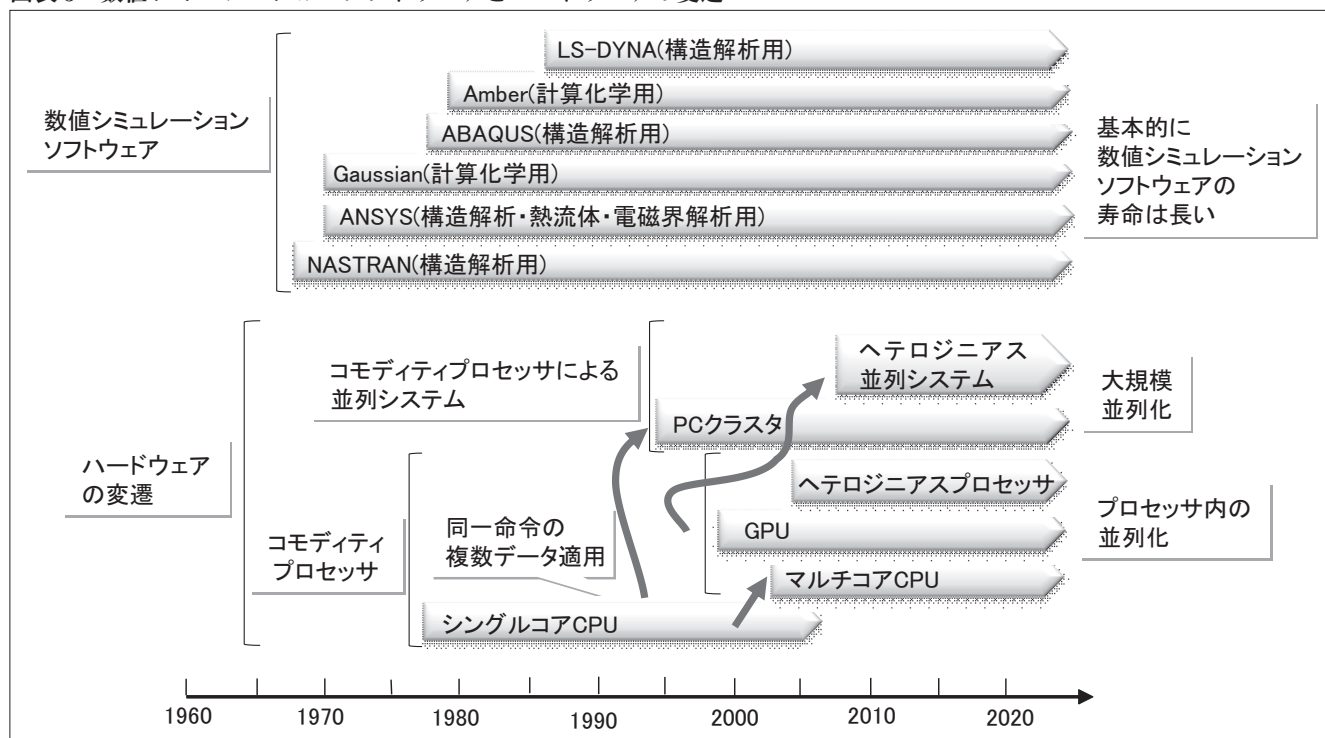
注1：GPUで扱うことのできる浮動小数点データが単精度から倍精度に拡張され、さらに、IEEE754浮動小数点演算標準に準拠したことで計算精度が要求される数値シミュレーションでも利用しやすくなった。  
 注2：IBM社のPowerXCell<sup>TM</sup>は、Cell/B.E.の倍精度浮動小数点演算を高速化した製品である。

図表2 ハードウェアアーキテクチャの変化と高性能ソフトウェア開発における難易度の増大



参考文献 9、20、26) を基に科学技術動向研究センターで作成

図表3 数値シミュレーションソフトウェアとハードウェアの変遷



参考文献 11~16) を基に科学技術動向研究センターで作成

能ソフトウェア開発は極めて難しい。並列分散システムでは、プロセッサ間のデータ分割・統合処理やデータ転送速度などの処理が実効効率に与える影響が大きくなり、これらを意識したソフトウェア開発が求められるからである。特に、機能の異なるコアが組み合わせられたヘテロジニアスプロセッサを用いた並列分散システムでは、各コ

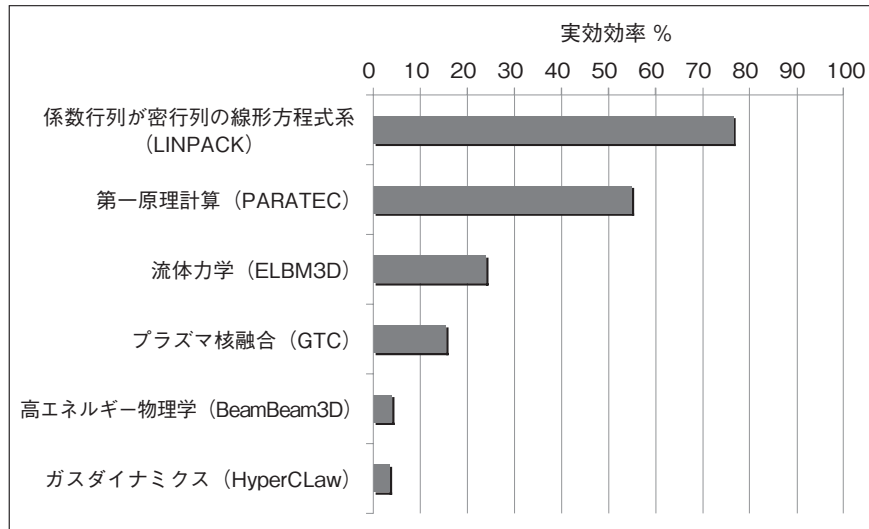
アの処理を効率的に連携させないと十分な性能が得られず、高性能ソフトウェア開発は一層難しいものとなっている。

## 2-2

### 数値シミュレーションソフトウェアの寿命

数値シミュレーションソフトウェアの寿命は、ハードウェアの寿命に比較すれば長いと言える。図表3には代表的な数値シミュレーションソフトウェアとコモディティプロセッサを中心としたハードウェアの変遷を示す。現在でも利用されている代表的な数値シミュレーションソフトウェアは約40年の歴史がある。一方、コモ

図表 4 並列分散システムにおける数値シミュレーションソフトウェアの実効効率 (AMD 社 Opteron プロセッサ 512 個を用いた並列分散システムによる計算結果)



参考文献<sup>5)</sup>を基に科学技術動向研究センターで作成

ディティプロセッサは短期間にアーキテクチャが変更され続けている(ここでは、これをハードウェアの寿命が短いとしている)。数値シミュレーションソフトウェアの寿命を延ばすために、言い換えると、同じソフトウェアをアーキテクチャが変更されたハードウェアでも使い続けるために、新たなハードウェアに合わせてソフトウェアをその都度修正しながら使用している。例えば、主な構造解析用や計算化学用の数値シミュレーションソフトウェアは、1960年代後半に登場し、機能拡張をしながら現在まで使用され続けている。

## 2-3

### 並列分散システムにおける数値シミュレーションソフトウェアの実効効率

並列分散システムでは、一般に数

値シミュレーションソフトウェアの実効効率が極めて低いことが知られている。実効効率とは理論性能を100%とした場合の実効性能の比である。図表4に、Olikerら<sup>5)</sup>が評価した数値シミュレーションソフトウェアとその実効効率の関係をまとめる。対象としたのはさまざまな科学技術領域から選択した以下に示す問題である。( )内に実効効率を調べるために使用された数値シミュレーションソフトウェアの名称を示す。

- 係数行列が密行列の線形方程式系 (LINPACK)
- 第一原理計算 (PARATEC)
- プラズマ核融合 (GTC)
- 流体力学 (ELBM3D)
- 高エネルギー物理学 (BeamBeam3D)
- ガスダイナミクス (HyperCFlow)

図表4に示した実効効率はAMD社のOpteronプロセッサ512個を用いた並列分散システムで得られ

た結果である。実効効率は、高性能コンピュータの性能測定に使用されるベンチマークプログラムであるLINPACKで70%以上になるが、PARATECを除いた数値シミュレーションソフトウェアの実効効率は25%にも満たないことが、図表4の例からわかる<sup>注3)</sup>。このように並列処理を導入しても数値シミュレーションソフトウェアの実効効率が低いことがあるのは、逐次処理を前提に作成されている場合、処理の依存関係により並列化されない部分が残るためである。さらに、仮に並列化が可能であっても、並列分散システムの場合には、プロセッサへのデータ転送の遅延や負荷分散の問題により実効効率を向上させることがさらに困難になる。このように、ハードウェアの高性能化に対して、数値シミュレーションソフトウェアの進歩が必ずしも追いついていないのが実情である。

注3: PARATECで実効効率が約55%と例外的に高いのは、計算の大部分を占めるFFT (Fast Fourier Transform) が並列処理により高速化されているためである。

### 3 数値シミュレーションの構成要素とソフトウェア基盤技術

数値シミュレーションの構成要素は非常に多くからなり、これらを整理すると、図表5に示す理論、数値モデル、アルゴリズム、ソフトウェア、ハードウェアの5階層として表される<sup>17)</sup>。

図表5の左側には従来の構成要素を、そして右側には新たに考慮すべき構成要素を示している。ここで、ソフトウェア基盤技術とは、ソフトウェア階層に分類される要素で、ハードウェアと連携する機能のように複数の数値シミュレーション間で共通的に利用されるものを指す。

図表左側の「従来の数値シミュレーションの構成要素」に示すように、従来からも各階層でさまざまな要素を考慮して数値シミュレーションソフトウェアの開発をする必要があったが、ソフトウェアが

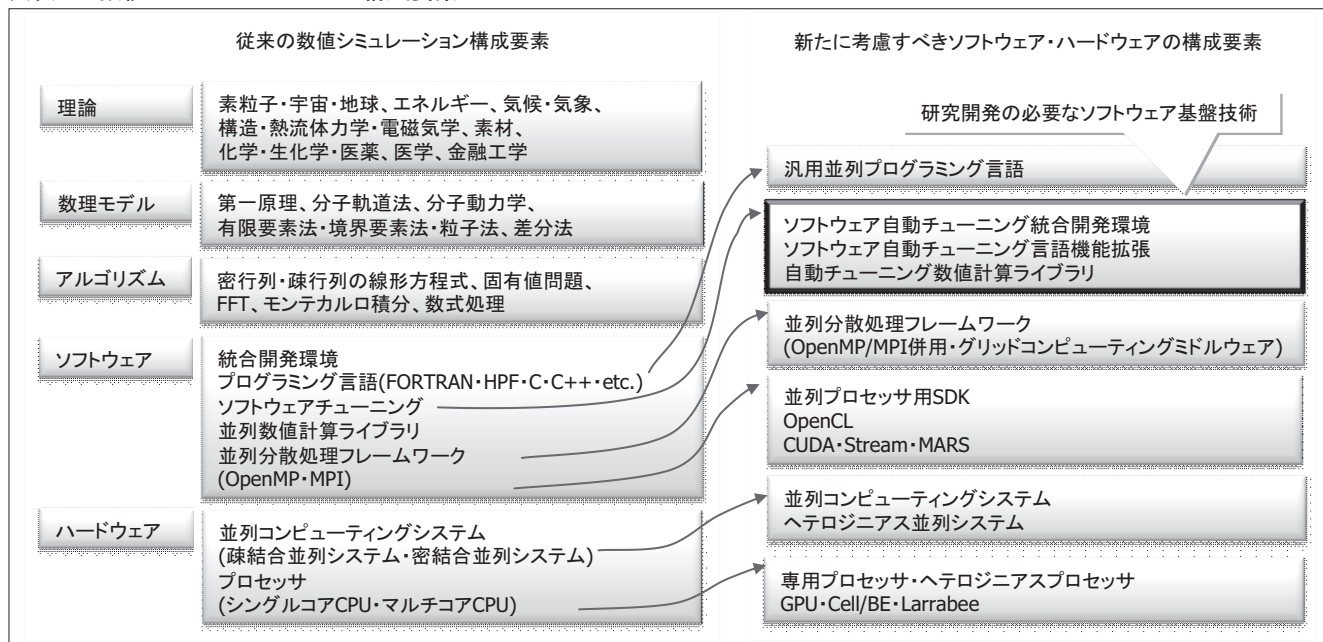
逐次処理中心の場合には、複雑な並列分散処理を行うプログラムを作成する必要がなかった。しかし、並列分散システム上で動く数値シミュレーションが求められるようになる、アルゴリズムはもちろん、場合によっては数理モデルや理論もハードウェアアーキテクチャを意識して考えなければならない状況になっている。つまり、かつては各階層の独立性が高かったが、並列分散システムの利用が進むことで各階層が密接に関係するよう変化してきている。

図表右側の「新たに考慮すべきソフトウェア・ハードウェアの構成要素」としては、ハードウェア階層での、GPU、ヘテロジニアスプロセッサ、これらを用いたヘテロジニアス並列システムの登場が挙げられる。ソフトウェアの階層では、

これらに対応した並列プロセッサ用の SDK (Software Development Kit) として CUDA<sup>20)</sup> や MARS<sup>24)</sup>、統一規格の OpenCL<sup>25)</sup> が加わる<sup>注4)</sup>。さらに、並列分散処理フレームワークに MPI・OpenMP の併用やグリッドコンピューティング用ミドルウェアも新たな要素として追加される。

新しいハードウェアの性能を引き出す高性能数値シミュレーションソフトウェアの開発では、これらの要素を考慮しなければならない。このような状況のため、高性能ソフトウェア開発においてソフトウェア自動チューニング関連技術が基盤技術として、より重要な役割を担うようになっている。これらについては、以降で詳述する。

図表5 数値シミュレーションの構成要素



参考文献 9, 17~25) を基に科学技術動向研究センターで作成

注4：Khronos Groupで策定しているマルチコアCPU, GPU, Cell/B.E.を統一的に扱うことのできるAPI(Application Program Interface)の仕様である。ただし、OpenCL 1.0では各種アーキテクチャに特化したプログラムを独自に記述する必要がある。

# 4 ソフトウェア自動チューニング

## 4-1

### 数値シミュレーションにおけるソフトウェア自動チューニング

ソフトウェアチューニングは、ハードウェアの性能を限界まで引き出すようにソフトウェアを調整する作業である。数値シミュレーションにおけるソフトウェア自動チューニングとは、ハードウェアに合わせたソフトウェアの調整を自動的に行い、手間のかかる手動チューニングをせずに実効効率を向上させることである。

図表6に、ソフトウェア自動チューニングの概要について例を挙げて説明する。例としては、偏微分方程式で記述される基礎方程式を有限要素法で解く場合を考える。この場合、数値シミュレーションでは線形方程式を解くことになる。ここで、計算速度や計算精度に影響を与える要因として図表6の下段に示す項目が存在し、これらがチューニング条件となる。チューニング条件は、シミュレーションモデル、線形方程式の数値

的性質、求解アルゴリズム、ソフトウェア品質、コンピュータシステムに分かれ、それぞれについて、さらに影響を与える要因がある。これらの要因を図表では矢印付きの記述として示している。

基本的なソフトウェア自動チューニングのプロセスは以下の5つのプロセスから成る。

- (1) 実験 チューニング条件を設定してソフトウェアを実行すること
- (2) 測定 実験結果から測定項目を求め評価関数を算出すること
- (3) 分析 測定項目、評価関数から性能モデルを推定すること
- (4) 学習 チューニング条件を自動修正すること
- (5) 決定 最適なチューニング条件を決定すること

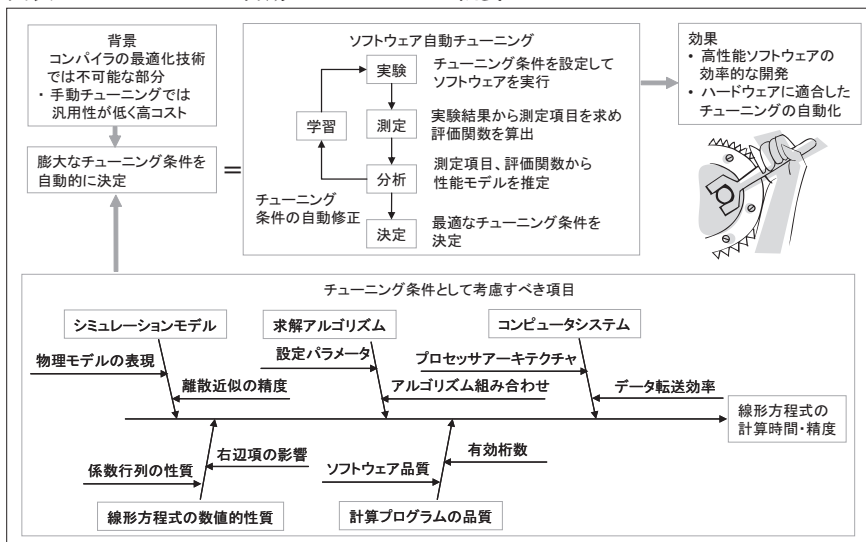
実際には(1)~(4)を繰り返してチューニング条件を変化させながら性能を向上させ、最終的に(5)で最適なチューニング条件を求める。なお、これらチューニング条件のうち、一部は従来のコンパイラの最適化技術によって自動化できる項目もあるが、アルゴリズム選択

やパラメータ調整までは対応ができない。そこで、数値シミュレーションの計算速度等の評価関数を設定して最適化問題を解くことにより最適なアルゴリズム選択やパラメータ調整といったチューニングの自動化が可能になる。

次に、ソフトウェアチューニングの効果と自動チューニングの必要性を、サイズが400×400の行列積計算を例にとり説明する。図表7のヒストグラムは、400×400の行列積計算で16,129通りの全チューニング条件について計算速度を求め、同一の計算速度となる結果をカウントしたものである。この場合、計算速度の最大値は1459MFLOPSであるが、ヒストグラムには1300, 1175, 1100MFLOPSに同一速度となるチューニング条件数のピークが存在し、計算速度を評価関数とした場合の最適解が容易には得られないことを示している。また、チューニング条件によって計算速度が大きく変化するため、最大計算速度の1/2にも満たない600MFLOPS付近にも小さなピークが存在する。探索空間をすべて調べれば最適解は得られるが、チューニングの計算コストが大きくなってしまふ。そこで、膨大なチューニング条件の候補から性能が最大化される条件を効率的に見つけ出す自動チューニング手法が必要になる。

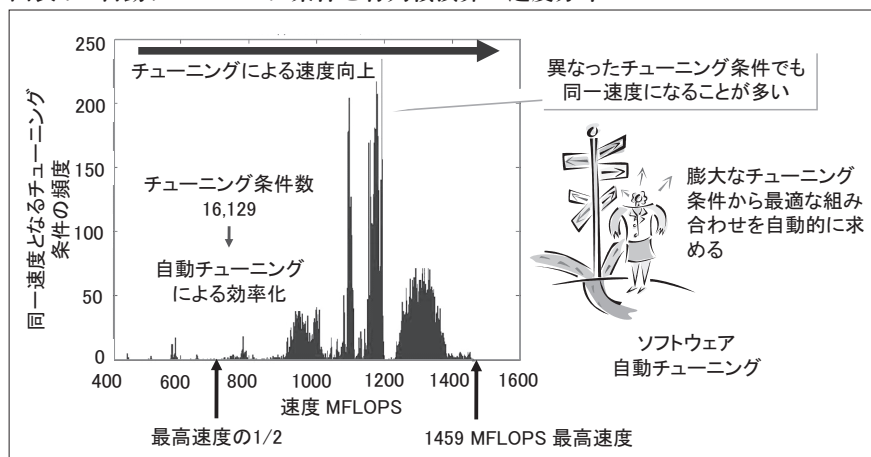
ソフトウェア自動チューニングは、静的な自動チューニング、動的な自動チューニング、高度な自動チューニングの機能に分けられる。自動チューニング機能付き数値計算ライブラリや自動チューニング機能付き数値シミュレーションソフトウェアはこれらの機能を用いて作成される。そして、自動チューニング機能の開発では、自

図表6 ソフトウェア自動チューニングの概要



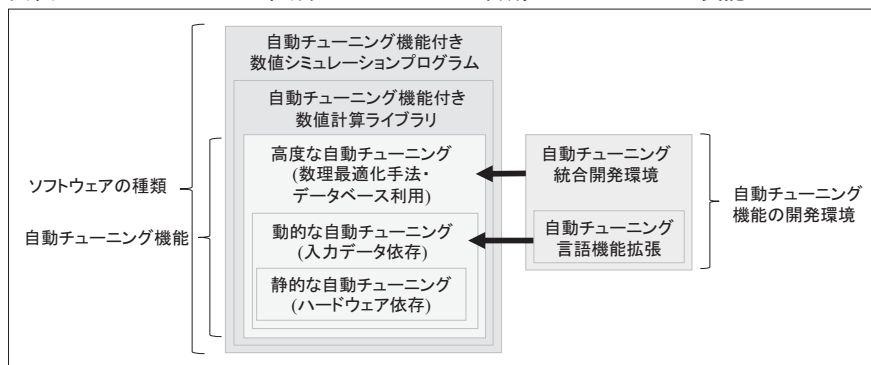
参考文献<sup>30~33)</sup>を基に科学技術動向研究センターで作成

図表7 自動チューニング条件と行列積演算の速度分布



参考文献<sup>40)</sup>を基に科学技術動向研究センターで作成

図表8 ソフトウェアの種類とソフトウェア自動チューニングの機能



参考文献<sup>30~33)、40、42)</sup>を基に科学技術動向研究センターで作成

動チューニング統合環境、自動チューニング言語拡張機能など適用する開発環境が異なる。これらの関係を図表8に示す。

過去の開発経緯を述べると、最初にソフトウェア自動チューニングが適用されたのは数値計算ライブラリのハードウェア依存部分であった。これは、現在、静的な自動チューニングと呼ばれている。次に、数値計算の入力データ依存性に注目して最適化を行うようになり、これを現在は動的な自動チューニングと呼んでいる。動的な自動チューニングには、静的なチューニング機能に加えて行列のサイズや非零要素の分布を調べて適切なチューニング条件を自動的に決定する機能

が含まれる。この動的な自動チューニング機能を実現するためには、開発環境としてプログラミング言語の機能拡張(言語機能拡張)を行う必要があった。さらに高度な自動チューニングを行うために、数理最適化手法やデータベースが利用されるようになり、現在も性能改善効果の検証を含めた自動チューニングソフトウェアのための統合開発環境を構築する研究が行われている。しかし、適用範囲は今のところ数値計算ライブラリに限定されており、今後は、行列計算や信号処理に帰結しない数値シミュレーションでも適用可能な自動チューニング研究の進展が期待されている。

## 4-2

### 自動チューニング機能付き数値計算ライブラリ

数値シミュレーションでは行列演算や信号処理の数値計算ライブラリの使用頻度が高く、これらの数値計算ライブラリに依存する計算が数値シミュレーションのボトルネックとなり性能を左右することが多い。そこで、線形方程式系・固有値問題・FFTなどの数値計算ライブラリに自動チューニング機能を組み込むことで、これらのライブラリを用いた数値シミュレーション性能は改善できる。

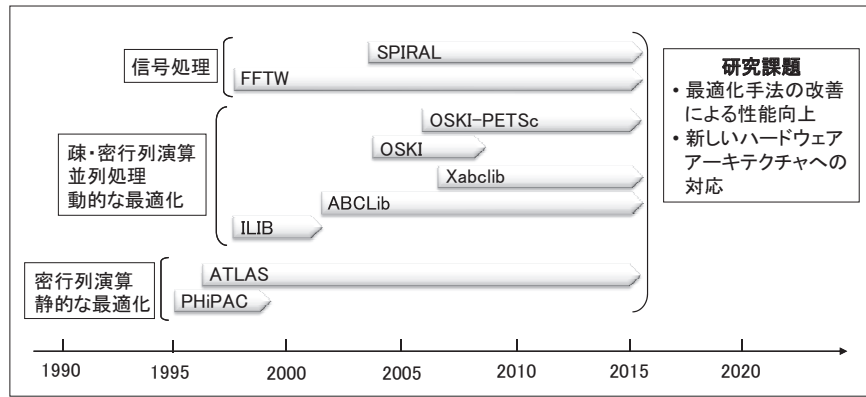
自動チューニング機能付き数値計算ライブラリも、ハードウェア依存部分を対象とした静的な自動チューニング機能付き数値計算ライブラリと、入力データに依存する部分を含めた動的な自動チューニング機能付き数値計算ライブラリに分類される。これら自動チューニング機能付き数値計算ライブラリのチューニング方法の特徴を以下に示す<sup>注5)</sup>。

- (a) 静的自動チューニング機能付き数値計算ライブラリ
  - インストール時にプロセッサのコア数やデータ転送速度などのハードウェア性能を評価して、性能が最大化されるようにライブラリの計算パラメータを調整する。
- (b) 動的自動チューニング機能付き数値計算ライブラリ
  - 行列のサイズや疎行列の非零要素の分布によって線形方程式や固有値問題のアルゴリズムや計算パラメータを選択する。

注5：線形方程式系と固有値計算問題では、アルゴリズムや使用される初期値やパラメータによっては反復計算で収束せず、解が得られないことがある。このような場合にも、適切な初期値やパラメータを決定する自動チューニング機能付き数値計算ライブラリは有用である。



図表9 自動チューニング機能付き数値計算ライブラリの開発動向



参考文献<sup>32, 33, 38, 39, 43)</sup>を基に科学技術動向研究センターで作成

図表10 研究開発されている自動チューニング機能付き数値計算ライブラリ

名称	種類	組織	機能
PHiPAC (Portable High Performance ANSI C)	密行列演算	米国 カリフォルニア大学 バークレー校	行列乗算ループをハードウェアアーキテクチャに合わせて自動的に高速化するライブラリ。コードジェネレータで複数のソースコードを生成し、処理速度の速いライブラリプログラムを選択する機能を実現している。具体的には、局所的な変数を導入してキャッシュによるメモリアクセス向上や、ループアンローリングによる依存関係のない部分の並列処理と分岐命令の排除により処理効率を改善する。
ATLAS (Automatically Tuned Linear Algebra Software)	密行列演算	米国 テネシー大学	BLAS(Basic Linear Algebra Subprograms)およびLAPACK (Linear Algebra PACKage)の一部を含む自動チューニング機能を有する行列演算ライブラリ。行列演算の性能に影響するブロックサイズやハードウェアに適したプログラムを複数用意しておき、ライブラリのインストール時に実行時間をタイマーで計測して適切なプログラムを選択する機能がある。
FFTW (the Fastest Fourier Transform in the West)	信号処理	米国 マサチューセッツ 工科大学	インストール時だけでなく実行時に入力データサイズに応じてメモリアクセス回数や演算量を少なくする自動チューニング機能をもつ高速フーリエ変換のライブラリ。現在ではMPIを用いた並列処理版やCell B.E.用のライブラリも開発されている。
SPiRAL (Software/Hardware Generation for DSP Algorithms)	信号処理	米国 カーネギーメロン 大学	FFT, DCT, Wavelet変換など信号処理用の自動チューニング機能を含むライブラリ。
OSKI (Optimized Sparse Kernel Interface)	疎行列演算	米国 カリフォルニア大学 バークレー校 ローレンス・リバモ ア国立研究所	BeBOP(Berkley Benchmarking and Optimization Group)で開発された疎行列用の自動チューニングライブラリ。MPI, BLASを用いた数値計算ライブラリPETScと連携することと並列処理にも対応している。
ILIB (Intelligent LIBrary)	密・疎行列演算	東京大学	係数行列が疎行列となる線形方程式の並列処理アルゴリズムが基になっており、非零要素の分布に応じて適したアルゴリズムを選択する機能をもつライブラリ。また、係数行列が密行列となる線形方程式系や固有値問題にも対応している。
ABCLib (Automatically Blocking and Communication-adjustment Library)	密・疎行列演算	電気通信大学	並列分散システム用の数値計算ライブラリ。キャッシュサイズのパラメータチューニングに相当する自動ブロック化と入力データに応じて自動的に最適なデータ通信方式を選択する機能がある。
Xabclib (eXtended ABCLib)	密・疎行列演算	東京大学	Open ATLibを用いてABCLibを拡張した自動チューニング並列処理ライブラリ。LANCZOS法による固有値問題やGMRES法による線形方程式系に対応している。

参考文献<sup>32, 33, 38, 39, 43)</sup>を基に科学技術動向研究センターで作成

動的に割り当てられるプロセッサやコア数に応じてチューニングを行う。これらの自動チューニング機能付き数値計算ライブラリの開発動向の歴史と今後予想される発展を図表9

にまとめる。PHiPAC, ATLASは静的な最適化を行う密行列演算用ライブラリである。信号処理用のFFTW, SPiRAL, 行列演算用のOSKI, ILIB, ABCLib, Xabclibは、並列分散システムに対応した動的

自動チューニング機能付き数値計算ライブラリである。ILIBが機能拡張されてABCLibになり、さらに、実行時の数値計算アルゴリズム選択と通信方式のチューニングが強化されてXabclibになってい

る<sup>33)</sup>。個々の自動チューニング機能付き数値計算ライブラリの開発動向を図表10に示す。図表10でわかるように多くの研究組織が、この自動チューニング機能付き数値計算ライブラリの研究に取り組んでいる。

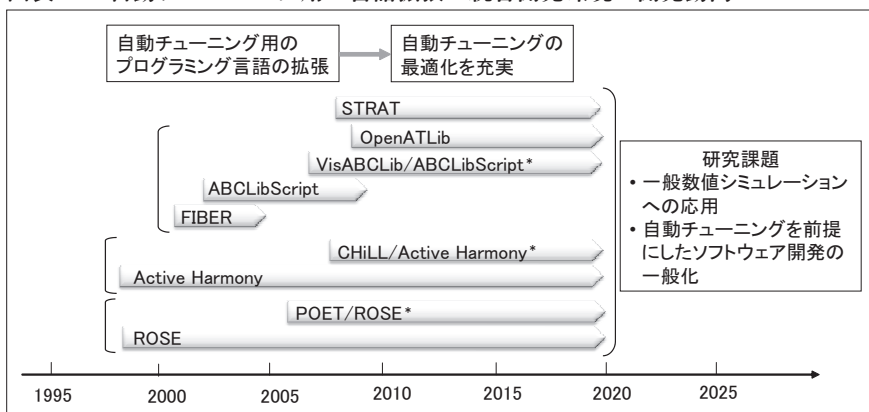
総じて、最適化手法の改善による性能向上、新しいハードウェアアーキテクチャにおける自動チューニングが今後の研究課題と言える。新しいアーキテクチャとして代表的なヘテロジニアス並列分散システムについては、人手によるチューニングですら研究段階にあり<sup>35、36)</sup>、今後の自動チューニングへの移行が期待される。

## 4-3

### 自動チューニングのための言語機能拡張・統合開発環境

前述したように、動的な自動チューニング機能を実現するために、既存のプログラミング言語に自動チューニング用の言語機能拡張が行われている。さらに、高度な自動チューニングのためには、性能評価や解析機能を含む自動チューニングの支援ツールが統合開発環境として利用されることになる。また、自動チューニング機能を含めたソースコードを作成しておけば、並列分散システムの規模が変化した場合にも、その変化に応じた性能改善が容易にでき、結果としてハードウェア依存性が抑えられてソフトウェアのポータビリティが向上するという利点もある。

図表11 自動チューニング用の言語拡張・統合開発環境の開発動向



参考文献<sup>31、34、41、42、44)</sup>を基に科学技術動向研究センターで作成

図表11にソフトウェア自動チューニングのための言語機能拡張と統合開発環境の開発の歴史と今後予想される開発動向<sup>31、33、40~43)</sup>をまとめる。ここで、\*のついたものが統合開発環境で、残りが言語機能拡張である。ROSE, Active Harmonyは各種プログラミング言語で記述されたソースコードに、自動チューニング用の性能計測機能を追加する機能をもつ言語機能拡張ツールである。POET, CHILLはそれぞれROSE, Active Harmonyと組み合わせることで、ソフトウェア自動チューニングの各種パラメータを最適化する機能を持つ統合開発環境である。ABCLibScriptは自動チューニング用の言語機能拡張で多機能であるが、対応プログラミング言語は現時点でFORTRANのみという問題があり、C言語対応が予定されている。VisABCLibはABCLibScriptに対応したソフトウェア自動チューニングの統合開発環境で、ソフトウェア性能の可視化機能が充実しているという特徴がある。また、SPRATはマルチコアCPU用とGPU用のCUDAのソ-

スコードを生成する機能をもつ言語機能拡張であり、ハードウェアの性能に応じて、自動的にCPU、GPUの計算を切り替えることで高性能化を実現している<sup>34)</sup>。

自動チューニング用の言語機能拡張・統合開発環境で高性能数値計算ライブラリが開発されているが、一般の数値シミュレーションで自動チューニング機能を適用するには行列演算や信号処理とは異なった課題が残る。例えば、流体と剛体や弾性体との相互作用や、分子間の動力的な相互作用などのシミュレーションでは、反復計算のループ内に条件分岐が存在することが多く、高性能数値計算ライブラリ用の自動チューニング手法では十分に対応できない。そこで、このような一般的なシミュレーションの自動チューニングを前提にしたソフトウェア開発が研究課題となっている。

図表12に示すように、言語機能拡張・統合開発環境についても、多くの研究機関で研究に取り組んでいる。

## 5 ソフトウェア自動チューニング応用に向けた新しい動き

ソフトウェア自動チューニングは、数値計算ライブラリの高性能化から研究が始められたという経

緯がある。そのため、線形方程式系や固有値問題など数値計算ライブラリの高性能化に重点が置かれ、

現状では、一般的な数値シミュレーションソフトウェアへの応用が積極的に行われているとは言い難い。

図表 12 研究開発されている自動チューニング言語機能拡張・統合開発環境の特徴

名称	組織	機能
ROSE	米国 ローレンス・リ バモア国立研 究所	Fortran, C, C++, OpenMP, UPCで記述されたソースコードを変換するための言語機能拡張。ROSEを用いることによって、各種プログラミング言語で記述されたソースコードに自動チューニング用の機能を実装することができる。
POET(Parameterized Optimizations for Empirical Tuning)	米国 テキサス大学 オースチン校	自動チューニングのパラメータ調整に、全サーチ、シンプレックス法、シミュレーテッドアニーリング、遺伝的アルゴリズムなどの適化手法を適用する統合開発環境。ROSEと組み合わせて利用する。
Active Harmony	米国 メリーランド大 学	ソフトウェア実行時にソフトウェアの性能を測定してフィードバックする機能などを含む自動チューニングの言語機能拡張。
CHILL	米国 南カリフォルニ ア大学	探索空間内の単体に相当する領域を変化させてパラメータ調整を行う最適化手法を導入した自動チューニングの統合開発環境。Active Harmonyと組み合わせて利用する。
FIBER (Framework of Install-time, Before Execute-time, and Run-time Auto-tuning)	東京大学	数値計算ライブラリのインストール時だけでなく、実行前、実行時の自動化まで含めた開発フレームワーク。以下の自動チューニングに対応している。 ・インストール時 ライブラリのインストールされるハードウェアに合わせた最適化 ・実行前 行列のサイズ等、問題に特化した最適化 ・実行時 疎行列の非零要素の分布を考慮した最適化、通信方式の最適化
ABCLibScript	電気通信大学	数値シミュレーションに特化した自動チューニング記述言語。ブロック幅調整、アルゴリズム選択、ループアンローリング調整の3つのチューニングを自動的に行う。FORTRANで記述されたソースコードにABCLibScript用の記述を追加して、ABCLibCodeGenによりソースコードを自動生成し、次に、性能サンプリングを繰り返すことでチューニングされたライブラリを自動生成する。
VizABCLib	電気通信大学	ABCLibScriptを用いたプログラム開発支援ツールであり以下の機能がある。 ・自動チューニングコードのインタラクティブ表示機能 ・自動チューニング過程のログ機能 ・性能予測値と実測値の比較機能 ・体系的性能評価機能 ・計算スキーム、アルゴリズムなど自動チューニングで必要される情報のデータベース。
SPRAT	東北大学	CPU, GPUに依存しないプログラミング言語のソースコードから、CPU用にはC++のソースコードを生成し、GPU用にはCUDAのソースコードを出力するコンパイラ。

参考文献<sup>31~34, 41~44)</sup>を基に科学技術動向研究センターで作成

一つの理由として、ソフトウェア自動チューニングを研究する計算機科学研究者と、各分野の数値シミュレーション研究者の連携が不十分であることが挙げられる。この連携を進める例として、米国エネルギー省科学局のSciDAC-2 (Scientific Discovery through Advanced Computing) というプログラムに、自動チューニングを含むソフトウェア性能工学を中心に研究を行っている PERI (Performance Engineering Research Institute)<sup>45)</sup> というプロジェクトがある。以下ではその活動内容を紹介する。

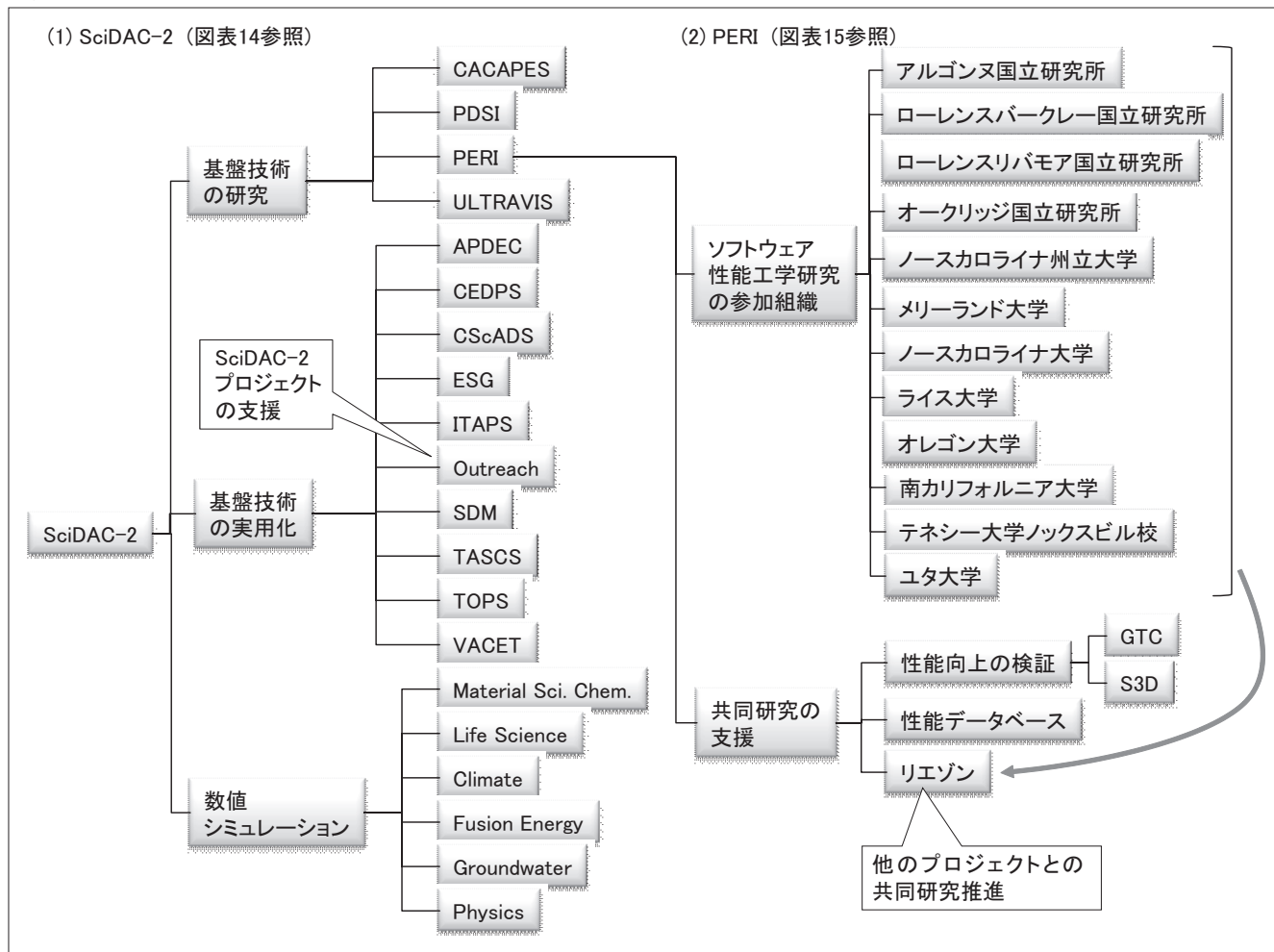
PERI が開始された背景は次の通りである。2001年に開始されたSciDAC-1というプログラムではPERIの前身にあたるPERC (Per-

formance Evaluation Research Center) というプロジェクトで、高性能数値シミュレーションプログラムのベンチマーク・解析と性能のモデル化・最適化研究を行い、その応用では気候予測モデル・プラズマ乱流・加速器シミュレーションを行っていた。これらの研究と応用から、2章で述べたような問題、すなわち、ソフトウェアに対してハードウェアの寿命が短く、新しいハードウェアに対応させるソフトウェア修正が研究を円滑に進める上で障害であることが明らかになった。また、数値シミュレーション研究者はソースコードの移植に関する情報を示さず、一方、計算機科学の研究者は既存ソフトウェアを移植するツールに興味を持たないという、それぞれの

研究者の問題も指摘された。このような問題を考慮しつつ、PERCの後継プロジェクトとしてPERIというプロジェクトが開始された。

現在、米国エネルギー省科学局のASCR (Advanced Scientific Computing Research) というプログラムにSciDAC、INCITE (Innovative and Novel Computational Impact on Theory and Experiment)の両プログラムが含まれる。SciDAC-2は数値シミュレーションのソフトウェア基盤技術に関する研究を行うプログラムであり、一方、INCITEは主に数値シミュレーションのための高性能ハードウェアと計算リソースを提供するプログラムである。

図表13 SciDAC-2プログラムとその中のPERIプロジェクトの構成



参考文献<sup>45)</sup>を基に科学技術動向研究センターで作成

## 5-1

### SciDAC-2における 数値シミュレーション 基盤ソフトウェアの研究

SciDAC-2は、図表13に示すように基盤技術の研究・基盤技術の実用化・数値シミュレーションを担当する3つの組織に大別される。図表14に2009年のSciDAC-2の各プロジェクトと研究テーマをまとめる。SciDAC-2内には、研究成果の公開などのOutreachセンターという支援組織も存在するのが特徴である。このOutreachセンターがプロジェクトの情報公開だけでなく、ユーザーサポートやトレーニングを担当し、プロジェクト間の連携を積極的に進める上で

重要な役割を果たしている。

## 5-2

### PERIというプログラムにおけるソフトウェア性能工学の研究

#### (1) PERIの役割

SciDAC-2プログラムの中で、他のプロジェクトで進められている数値シミュレーション研究に高性能ソフトウェア開発技術を提供することがPERI(ソフトウェア性能工学研究所)の目的である。PERIは、以下の研究開発を担当している。

- 数値シミュレーションの性能モデル化  
開発したソフトウェアで得られ

る処理速度を正確に予測する。

- ソフトウェア自動チューニングの研究開発  
研究者が行うプログラミングの負担を軽減するという難易度が高い長期的な研究目標を立てている。
- アプリケーション連携  
PERIの研究成果をSciDAC-2内の他プロジェクトが研究開発している数値シミュレーションに応用する。

#### (2) PERIの組織と運営

PERIの組織と運営に着目すると、その特徴は、図表5に示した数値シミュレーションの構成要素を包括的に捉え、ソフトウェア基盤技術におけるソフトウェア性能工学の研究開発を中心にして、生産性の高い組織を構成し運営して

図表 14 SciDAC-2 プログラムに含まれる組織と課題

	組織の略称	組織の名称(日本語)	課題
基盤技術の研究	CACAPES	組み合わせ科学計算とベタスケールシミュレーション研究所	並列計算の負荷分散、自動微分、疎行列計算
	PDSI	ペタスケールデータストレージ研究所	データストレージを中心に仕様、標準規格、アルゴリズム、性能計測ツール開発
	PERI	ソフトウェア性能工学研究所	ソフトウェア性能モデル化と性能予測、ソフトウェア自動チューニング、アプリケーションを対象としたソフトウェア性能工学
	ULTRAVIS	超スケール可視化研究所	膨大なデータ集合から潜在的な情報を抽出する可視化ツールの開発
基盤技術の実用化	APDEC	応用偏微分方程式センター	偏微分方程式のためのアルゴリズムとソフトウェアフレームワーク
	CEDPS	分散ペタスケール科学センター	高信頼性、高性能なデータ転送機構とグリッドにおけるリソース割り当てと仮想化環境
	CScADS	スケーラブルアプリケーション開発ソフトウェアセンター	ペタスケール計算プラットフォーム、通信ライブラリ、数学ライブラリ、オープンソースコンパイラ
	ESG	地球システムグリッドセンター	気候、気象予測のための大気、海洋、陸が統合された次世代シミュレーションのデータ生成
	ITAPS	先端ペタスケールシミュレーションのための共同利用技術センター	SciDACアプリケーションで相互利用、あるいは、互換性のあるメッシュ、ジオメトリ等のデータ操作ツール
	Outreach	アウトリーチセンター	プロジェクト間の情報提供やサポートサービス、トレーニング、新しい組織へのSciDAC研究成果の移転
	SDM	科学データ管理センター	科学技術計算ワークフローの自動化、データマイニング、データ解析、ストレージへの効率的なアクセス
	TASCS	先端科学コンポーネントソフトウェア技術センター	並列シミュレーション、ハードウェア、ソフトウェアの品質、頑健性、動的適合性、ユーザビリティ向上を目的としたコンポーネントソフトウェア化
	TOPS	ペタスケール最適シミュレーション	スケーラブルソルバーとアプリケーションのボトルネック解消のための研究開発
	VACET	可視化・解析センター	データの可視化、解析ソフトウェア研究開発
数値シミュレーション	Material Sci. Chem.	素材科学・化学	ペタスケール計算化学・素材モデリング、ナノ構造の量子シミュレーション、応力下の亀裂解析、流体内における化学反応と相互作用
	Life Science	微生物・細菌が環境に与える影響とエネルギー生成	微生物による水素生成、バイオエタノール、エネルギー生成
	Climate	気候・気象	地球気候の物理的・化学的モデル、雲のモデル化と検証、大域気候モデルの改善、大気モデル計算の連続的・動的かつ適応的なグリッドコンピューティング
	Fusion Energy	代替クリーンエネルギー	プラズマ核融合のための乱流解析
	Groundwater	地下水	汚染物質の地下水による拡散、地表化における地形・生物・化学モデル
	Physics	物理	素粒子、核エネルギー、宇宙物理学、衝撃波の乱流解析、オープンサイエンスグリッド

参考文献<sup>45)</sup>を基に科学技術動向研究センターで作成

いる点にある。具体的には、研究開発の目的を共有し、個々の基礎研究と応用研究開発を有機的に機能させることで、結果として実用化までの障害を早期に取り除くという点で優れていると考えられる。

図表 13 に示したように、PERI の中にはソフトウェア性能工学の研究を進めるグループと数値シミュレーション応用などの共同研究を支援するグループが含まれている。図表 15 には、その両方を同時に行う 4 つの国立研究所と 8 大学の担当課題を示す。各プロジェクトの研究開発内容だけでなく、PERI 内の組織、SciDAC-2、INCITE な

ど他のプロジェクトとの連携を含めた活動が決められているのが特徴と言える。共同研究を支援する組織にはプラズマ核融合シミュレーションへの応用(GTC)、流体シミュレーション(S3D)、性能データベース構築、他のプロジェクトとの共同研究における窓口となるリエゾンも含まれる。ただし、リエゾンはソフトウェア性能工学の研究を進めるグループのメンバーと重複した体制をとっている。

PERI の全体会議は毎年開催し、隔週で電話会議を行うことで連携を緊密にし、予定外の会議は月曜の午前に開催するといった取り決

めも公開されている。さらに、限られたリソースを重要な SciDAC-2 プロジェクトに集中させ、ソフトウェア性能工学とは関係のない一般の計算機科学や数学の研究組織とならないように注意が払われている。このように、効率的な研究マネジメントが行われ、さらに、組織としては SciDAC-2 プロジェクトの支援を行う Outreach センター、および、PERI 内のリエゾンが、研究成果の数値シミュレーション応用を円滑に進める役割を担っている。

図表 15 PERI の参加組織と担当する課題

組織	担当する課題
アルゴンヌ国立研究所	数値シミュレーションのためのサービスインフラストラクチャソフトウェア品質向上(TASCSと連携) ソフトウェア性能データベースのインタフェース拡張とアプリケーション開発者用の付加的な簡易インタフェースの追加 自動チューニングの学習を自動化するためのインターフェースの定義とコンポーネント実装 ハードウェア性能データベースを各アプリケーションで共有するインフラストラクチャ PerfExploereを用いた性能解析インターフェースの拡張 機械学習に基づく解析コンポーネントプロトタイプのプロバスタ化(オレゴン大学との連携) アルゴンヌ国立研究所のBlue Gene/Pとオークリッジ国立研究所のCray XT3で、FLASHアプリケーションの性能評価
ローレンスバークレー国立研究所	PERIプロジェクト管理 PERI、および、基盤技術の研究組織の進捗確認 プラズマ乱流解析チームの解析処理 新規太陽電池のための素材シミュレーション用の量子計算ソフトウェアのチューニング アプリケーションでのマルチコアプロセス用自動チューニング機能の開発とテスト
ローレンスリバモア国立研究所	自動チューニングのためのツールであるPOETを利用した経験的なチューニング ソフトウェア性能評価、特にクロスプラットフォームモデルの調査継続 他組織のPERI研究者と連携して各種性能予測ツールを統合 ソフトウェア自動チューニング、性能予測ツールのSciDACアプリケーション適用 MPIアプリケーションの挙動を示すモデル生成 通信パターンを計測するMPIレーシングメカニズムの拡張 MPIイベントの性能分布を計測する機能の実装 タイムスタンプを含み理想的なトレースを行うための先進的手法の調査 性能向上検証チームの活動促進
オークリッジ国立研究所	性能向上を検証するチームで必要とされるネットワークポロジィやルーティング設定のインターコネクトシミュレータ開発の継続 大規模システムでの性能計測のためのモデルやシミュレーション結果の比較 スケールリングボトルネックを特定するモデルの高精度化 気候・気象、核融合、マテリアルサイエンス、地下水アプリケーションの共同研究サポート PERIの研究成果応用の促進 SciDAC外プロジェクトとの連携推進
ノースカロライナ州立大学	アプリケーションチームの共同研究サポートの継続し、Cray XT4, BlueGene/Pでの性能解析と最適化を行う アプリケーションチームとPETSc開発者、利用者とのコミュニケーションを継続し、I/O、ユーザールーチンを改善
ライス大学	OSのハードウェアカウンタ機能を用いた性能サンプリングの問題を解決するためにCray, IBMとの共同作業を継続 OSの問題が解決した段階でHPCToolkitを導入 HPCToolkitの最適化コードのパスプロファイリング機能の拡張を継続 OpenMP, MPI+OpenMPプログラム用性能解析手法の拡張を継続 SciDAC, INCITEアプリケーションチームとの連携
カリフォルニア大学サンディエゴ校	ネットワークシミュレータの $\alpha$ テスト ネットワークシミュレータの $\beta$ リリース 大規模データ、プロセスシステムでのメモリトレース推定のための基本的な研究開発
メリーランド大学	経験的な検索機能を含む自動チューニングフレームワークの統合 Active Harmony と CHiLLフレームワークの統合を完了し評価を開始 PERI-DB検索APIの開発 性能向上検証チームのサポート
オレゴン大学	ベタスケールアプリケーションの性能計測と解析のサポート継続 性能向上検証チームの流体解析、プラズマ乱流解析アプリケーションの性能計測と解析の継続 PERI-DBグループと性能データベースの統合を継続 性能向上の検証チームのデータの解析にPerfExplorerを利用
南カリフォルニア大学	PERIプロジェクト全体の管理 自動チューニング利用者のためのAPI開発の継続 自動チューニングとCHiLLに関係する要求仕様を決めるための研究 SciDAC, INCITEアプリケーションチームとの連携 データコピーライブラリの研究開発(ユタ大学と連携) Active Harmony と CHiLLの統合を継続(メリーランド大学、ユタ大学との共同研究)
テネシー大学ノックスビル校	PERI性能モデリング、自動チューニングをサポートするクロスプラットフォームの性能カウンタライブラリの開発を継続 自動チューニングのための経験的な探索手法の研究とPERI自動チューニングフレームワークへの統合 マルチコアアーキテクチャのための最適化手法の研究とPERI自動チューニングフレームワークへの統合 性能向上の検証チームのデータベース構築に協力
ユタ大学	PERIの自動チューニンググループの牽引し、外部の共同研究者と報告書作成、ポスター発表、論文発表を調整 自動チューニングコンパイラ技術にスレッドメカニズムを導入 データコピーライブラリの開発(南カリフォルニア大学と共同研究) Active HarmonyとCHiLLの統合を継続(南カリフォルニア大学、メリーランド大学との連携) CHiLLの安定版リリースのための作業(南カリフォルニア大学と連携)

参考文献<sup>46)</sup>を基に科学技術動向研究センターで作成

## 6 日本国内における研究推進のための課題

これまで述べたように、コモディティプロセッサを用いた並列分散システム上での数値シミュレーションが増えつつあるが、これらのハードウェアの性能を活かす高性能ソフトウェア開発が極めて難しくなっている。そのため、今後の高性能ソフトウェア開発では自動チューニング技術の中核にしたソフトウェア基盤技術が重要な役割を担っている。特にヘテロジニアス並列分散システムについては自動化以前のチューニング自体が研究段階にあり、自動チューニングの実用化を目指した研究を進める必要がある。

日本国内でも数値計算を専門とする大学、および、企業の研究者を中心に、自動チューニング研究会が発足しており、自動チューニング技術に関する課題調査の報告<sup>48)</sup>と自

動チューニング記述言語 OpenATLib での API (Application Program Interface) の仕様策定を行っている。また、2006 年以降に自動チューニング研究会が開催している国際ワークショップ iWAPT (International Workshop on Automatic Performance Tuning) は海外の研究者からも注目されている。

このように日本国内では米国と比較しても遜色のない研究が行われていると思われるが、その推進体制には課題が残る。日本では、研究者を中心に組織が組まれているため、研究から最終的な数値シミュレーションへの応用、実用化までのロードマップ作成や研究者間での役割分担やリソース共有を十分に行えていない。特に大学や公的研究機関で行うソフトウェア基盤技術の研究開発は、米国の

SciDAC-2 やその中の PERI のように関連する応用プロジェクトを俯瞰した上で、研究と応用を並行して進めるのが最も効率がよいと考えられる。今後、日本国内の研究組織には、基盤技術研究と応用研究を効率的に行うための研究マネージメントの再考を期待したい。

### 謝辞

東京大学情報基盤センター 中島研吾 教授、片桐 孝洋 特任准教授、(独)理化学研究所 情報基盤センター 伊藤 祥司氏、東京工業大学 情報理工学研究所 遠藤 敏夫 特任准教授、(独)海洋研究開発機構 渡邊 國彦センター長、(株)フィックスターズ 三木 聡 CEO、田村陽介 CTO から、本稿をまとめるにあたり貴重な意見をいただきました、ここに深謝致します。

### 略語表

ABCLib	: Automatically Blocking and Communication-adjustment Library
APDEC	: Applied Partial Differential Equations Center
ASCR	: Advanced Scientific Computing Research
ATLAS	: Automatically Tuned Linear Algebra Software
CACAPES	: Combinatorial Scientific Computing and Petascale Simulations Institute
CEDPS	: Center for Enabling Distributed Petascale Science
CScADS	: Center for Scalable Application Development Software
ESG	: Earth System Grid Center for Enabled Technologies
FFT	: Fast Fourier Transform
FFTW	: the Fastest Fourier Transform in the West
FIBER	: Framework of Install-time Before Execute-time, and Run-time auto-tuning
GPU	: Graphics Processing Unit
GTC	: Gyrokinetic Turbulence Code
HPC	: High Performance Computing
ILIB	: Intelligent Library
INCITE	: Innovative and Novel Computational Impact on Theory and Experiment
ITAPS	: Interoperable Technologies for Advanced Petascale Simulations Center
iWapt	: International Workshop on Automatic Performance Tuning
OSKI	: Optimized Sparse Kernel Interface
PARATEC	: Parallel Total Energy Code

PDSI : Petascale Data Storage Institute  
PHiPAC : Portable High Performance ANSI C  
PERC : Performance Evaluation Research Center  
PERI : Performance Engineering Research Institute  
SciDAC : Scientific Discovery through Advanced Computing  
SDK : Software Development Kit  
SDM : Scientific Data Management Center  
SPIRAL : Software/Hardware Generation for DSP algorithms  
TASCS : Center for Technology for Advanced Scientific Component Software  
TOPS : Towards Optimal Petascale Simulations  
ULTRAVIS : Institute for Ultra-Scale Visualization  
VACET : Visualization and Analytics Center for Enabling Technologies  
Xabclib : eXtended ABCLib

## 参考文献

- 1) World Technology Evaluation Center, Inc., WTEC Report on International Assessment of Research and Development in Simulation-Based Engineering and Science, (Apr. 2009)
- 2) 野村稔、米国政府の高性能コンピューティングへの取り組み、科学技術動向、No. 47、(2005.02)
- 3) 館山佳尚、ナノテクノロジー開発の促進に向けたナノシミュレーション技術の普及、科学技術動向、No. 59、(2006.02)
- 4) 野村稔、欧州におけるベタスケールコンピューティングの動向、科学技術動向、No. 79、(2008.10)
- 5) Leonid Oliker, Andrew Cannig, Jonathan Carter, Costin Iancu, Michael Lijewski, Shoaib Kamil, John Shalf, Hangzhang Shan, Eric Strohmaier, Stephan Ethier, Tom Godate, Scientific Application Performance on Candidate PetaScale Plathome, Proc. IPDSP, (2007)
- 6) 佐藤文俊、平野敏行、阿部俊彦、上村典子、恒川直樹、西村康幸、山口 智美、湯川英宜、石川健太郎、千葉貢治、密度汎関数法によるタンパク質の全電子シミュレーション、第28回日本シミュレーション学会大会発表講演論文集、pp. 163-166、(2009.06)
- 7) 独立行政法人海洋研究開発機構、地球シミュレータ産業利用シンポジウム発表資料集、(2008.09)
- 8) 高性能コンピュータへの搭載プロセッサ数の増加、科学技術動向、No. 90、(2008.09)
- 9) IBM, Cell Broadband Engine Technology  
[http://www-03.ibm.com/technology/resources/technology\\_cell\\_pdf\\_CellBrBandEngineWhitePaper.pdf](http://www-03.ibm.com/technology/resources/technology_cell_pdf_CellBrBandEngineWhitePaper.pdf)
- 10) Richard Walsh, Steve Conway, Earl C. Joseph, Jie Wu, With Its New PowerXCell 8i Product Line, IBM Intends Take Accelerated Processing into the HPC Mainstream, August 2008  
[http://www-03.ibm.com/technology/resources/technology\\_cell\\_IDC\\_report\\_on\\_PowerXCell.pdf](http://www-03.ibm.com/technology/resources/technology_cell_IDC_report_on_PowerXCell.pdf)
- 11) MSC NASTRAN, <http://www.mscsoftware.co.jp/products/nastran/>
- 12) ANSYS Inc., <http://www.ansys.com/>
- 13) The Official Gaussian Website, <http://www.gaussian.com/>
- 14) SIMULIA, <http://www.simulia.com/>
- 15) Amber and NPACI : A Strategic Application Collaboration for Molecular Dynamics  
[http://www.sdsc.edu/pub/envision/v14.4/sac\\_amber.html](http://www.sdsc.edu/pub/envision/v14.4/sac_amber.html)
- 16) Livermore Software Technology Group, <http://www.lstc.com>
- 17) 中島研吾、東京大学における全学的HPC教育の試み「学際計算科学・工学人材育成プログラム」  
<http://hss.iic.hokudai.ac.jp/WS07/pdf/Nakajima.pdf>
- 18) OpenMP, <http://openmp.org/wp/>
- 19) MPICH2, <http://www.mcs.anl.gov/research/projects/mpich2/>
- 20) CUDA Zone, [http://www.nvidia.com/object/cuda\\_home.html#](http://www.nvidia.com/object/cuda_home.html#)



- 21) ATI Stream Software Development Kit  
<http://developer.amd.com/gpu/ATIStreamSDK/Pages/default.aspx>
- 22) General-Purpose Computation on Graphics Hardware, <http://gpgpu.org/>
- 23) Aleksey Bader et al., Game Physics Performance on the Larrabee Architecture  
[http://download.intel.com/technology/architecture-silicon/GamePhysicsOnLarrabee\\_paper.pdf](http://download.intel.com/technology/architecture-silicon/GamePhysicsOnLarrabee_paper.pdf)
- 24) MARS : Multicore Application Runtime System  
<http://ftp.uk.linux.org/pub/linux/Sony-PS3/mars/1.1.4/mars-docs-1.1.4/html/>
- 25) The Khronos OpenCL Working, The OpenCL Specification, Version 1.0, Document Revision 29  
[http://www.khronos.org/news/press/releases/the\\_khronos\\_group\\_releases\\_opencl\\_1.0\\_specification/](http://www.khronos.org/news/press/releases/the_khronos_group_releases_opencl_1.0_specification/)
- 26) 遠藤敏夫、東京工業大学 TSUBAME における専用計算ユニット活用事例、情報処理、Vol. 50、No. 2、pp. 100-106、(2009.02)
- 27) 青木尊之、フル GPU による CFD アプリケーション、情報処理、Vol. 50、No. 2、pp. 107-115、(2009.02)
- 28) 塚本明、安田絹子、田村陽介、町田浩之、Cell/B.E. プログラミングの特徴と活用事例の紹介、情報処理、Vol. 50、No. 2、pp. 116-128、(2009.02)
- 29) 成見哲、濱田剛、小西史一、専用計算ユニットによる粒子法シミュレーションの加速、情報処理、Vol. 50、No. 2、pp. 129-139、(2009.02)
- 30) 須田 礼仁、ソフトウェア自動チューニングの数理、情報処理、Vol. 50、No. 6、pp. 487-479、(2009.06)
- 31) 伊藤 祥司、ソフトウェア自動チューニングのための支援ツール、Vol. 50、No. 6、pp. 499-504、(2009.06)
- 32) 黒田久泰、直野健、岩下武史、自動チューニング機能付き数値計算ライブラリ、情報処理、Vol. 50、No. 6、pp. 505-511、(2009.06)
- 33) 片桐孝洋、ソフトウェア自動チューニング記述のための計算機言語、情報処理、Vol. 50、No. 6、pp. 494-498、(2009.06)
- 34) 滝沢寛之、GPU コンピューティングにおけるソフトウェア自動チューニング、情報処理、Vol. 50、No. 6、pp. 527-531、(2009.06)
- 35) 霜田善道、PowerXCell と線形計算、先駆的科学計算に関するフォーラム 2008 ～線形計算を中心に～、(2008.09)
- 36) 遠藤敏夫、額田彰、松岡聡、丸山直也、實本英之、四種プロセッサからなるヘテロ型スーパーコンピュータにおける Linpack チューニング、第 16 回「ハイパフォーマンスコンピューティングとアーキテクチャの評価」に関する北海道ワークショップ (HOKKE-2009)、(2009.02)
- 37) 片桐孝洋、櫻井隆雄、黒田久泰、直野健、中島研吾、OpenATLib : 汎用的な自動チューニングインターフェースの設計と実装、SwoPP2009、(2009.08)
- 38) Richard Vuduc, James W. Demmel, and Katherine A Yelick, OSKI : A library of automatically tuned sparse matrix kernel, Journal of Physics : Conference Series, Vol. 16, No. 1. 512 (2005)
- 39) M. Puschel et al., SPIRAL : A Generator for Platform-Adapted Libraries of Signal Processing Algorithm, International Journal of High Performance Computing Applications, Vol. 18, No. 1, 21-45 (2004)
- 40) Keith Seymour, Haihang You, Jack Dongarra, A Comparison of Search Heuristics for Empirical Code Optimization, Proc.2008 IEEE International Conference on Cluster Computing, pp. 421-429, (Oct. 2008)
- 41) ROSE, <http://www.rosecompiler.org/>
- 42) Qu Yi, Keith Seymour, Haihang You, Richard Vuduc, Dan Quinlan, POET : Parameterized Optimization for Empirical Tuning, Proc. IPDPS 2007, pp. 1-8, (Mar. 2007)
- 43) PETSc, <http://www.mcs.anl.gov/petsc/petsc-as/>
- 44) Ananta Tiwari, Chun Chen, Jacqueline Chame, Mary Hall, Jefferey K. Hollingsworth, A Scalable Autotuning Framework for Compiler Optimization, Proc. IPFPS 2009, (May. 2009)
- 45) SciDAC, <http://www.scidac.gov/>
- 46) Performance Engineering Research Institute, <http://www.peri-scidac.org/perci/>
- 47) The Office of Advanced Scientific Computing Research  
<http://www.er.doe.gov/ascr/About/about.html>
- 48) 自動チューニング研究会、自動チューニング技術に関する課題調査、(2008.11)

---

## 執筆者プロフィール

---



### 古川 貴雄

推進分野ユニット  
科学技術動向研究センター 上席研究官  
<http://www.nistep.go.jp/index-j.html>

ITベンチャー企業でコンピュータグラフィックスを用いた設計支援システム、実時間動画像処理を応用したアプリケーションの研究開発に従事し、2009年より現職。



### 野村 稔

情報・通信ユニット  
科学技術動向研究センター 客員研究官  
<http://www.nistep.go.jp/index-j.html>

企業にてコンピュータ設計用CADの研究開発、ハイ・パフォーマンス・コンピューティング領域、ユビキタス領域のビジネス開発に従事後、現職。スーパーコンピュータ、LSI設計技術等、情報通信分野での科学技術動向に興味を持つ。