

式が書ければ「京」が使える

ー 高度なプログラムを自動生成できる新言語「Formura」を開発ー

要旨

理化学研究所（理研）計算科学研究機構コデザイン推進チームの村主崇行特別研究員らと、千葉大学の堀田英之特任助教、神戸大学の牧野淳一郎教授、京都大学の細野七月特任助教、富士通株式会社の井上晃マネージャーらの共同研究グループ^{*}は、スーパーコンピュータ「京（けい）」^[1]を用いて、数式のような簡潔な指示を書くだけでスーパーコンピュータでの計算に必要な高度なプログラムを自動生成できるプログラミング言語「Formura」を開発しました。

スーパーコンピュータでの計算に必要なプログラムはときに数十万行にも及び、作成やチューニングは大変困難です。一方で、原理的にはシミュレーションしたい自然現象とその離散化法^[2]を指定すれば、プログラムは機械的に生成できます。しかし、プログラミングはシミュレーションとコンピュータ双方に深い知識が必要となる非常に高度な作業であり、多数の計算機を協調して動作させるスーパーコンピュータの性能を引き出す高度なプログラムを、自動かつ汎用的に生成することは不可能でした。

そこで共同研究グループは、方程式がプログラムに変換されるまでの一連の段階に対応する数学的定義を作りました。スーパーコンピュータが持つ階層のすべての段階において、自然が元来備えている「並列性」と「局所性」^[3]を保持する変換を厳密に定めることで、新たなプログラミング言語「Formura」を開発しました。これによって、これまで不可能だったプログラミングの機械化に成功しました。さらにFormuraは、同じアプリケーションに対して何万通りものプログラムを試し、最も速かったものを自動的に選択します。

Formuraを開発したことで、規則格子シミュレーション^[4]分野においては、自然科学者が慣れ親しんだ方程式の記法を使ってシミュレーションしたい対象を記述することで、スーパーコンピュータの性能を引き出すための高度なプログラムが自動的に作成できるようになりました。気象、地震、宇宙、生態ネットワークの研究など、規則格子シミュレーションを用いる分野の研究の加速が期待できます。

本研究成果は関数型言語を扱う科学誌の高性能計算部門『*FHPC 2016: Proceedings of the 5th International Workshop on Functional High-Performance Computing*』（9月22日付）および高性能計算技術を扱う科学誌『*SC '16 Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis Article No. 3*』に掲載され、高性能計算技術の国際的な賞であるゴードン・ベル賞^[5]のファイナリストに選ばれました。

※共同研究グループ

理化学研究所 計算科学研究機構

フラッグシップ2020プロジェクト

コデザイン推進チーム

特別研究員	村主 崇行	(むらぬし たかゆき)
研究員	似鳥 啓吾	(にたどり けいご)
特別研究員	岩澤 全規	(いわさわ まさき)
研究員	丸山 豊	(まるやま ゆたか)

研究部門

複合系気候科学研究チーム

チームリーダー	富田 浩文	(とみた ひろふみ)
研究員	西澤 誠也	(にしざわ せいや)
研究員	八代 尚	(やしろ ひさし)

連続系場の理論研究チーム

研究員	中村 宜文	(なかむら よしふみ)
-----	-------	-------------

千葉大学 大学院理学研究科 基盤理学専攻 物理学コース

特任助教	堀田 英之	(ほった ひでゆき)
------	-------	------------

神戸大学 大学院理学研究科 惑星学専攻 新領域惑星学講座

教授	牧野 淳一郎	(まきの じゅんいちろう)
----	--------	---------------

(理研計算科学研究機構コデザイン推進チーム チームリーダー)

京都大学大学院 総合生存学館

特任助教	細野 七月	(ほその なつき)
------	-------	-----------

富士通株式会社 TC ソリューション事業本部 TC フロンティアセンター

マネージャー	井上 晃	(いのうえ ひかる)
--------	------	------------

1. 背景

今日のシミュレーション科学を支えるスーパーコンピュータは、多数の計算機が高速ネットワークで接続された構造をしており、機種によっては数十万から数億もの計算を並列に実行する能力を持ちます。しかし、それら計算機の全てに適切に計算を割り振って、タイミングをあわせ協調して動作させない限り、性能を完全に引き出すことはできません。もともと、高精度シミュレーションのためのプログラムは、基礎的な方程式からいくつもの段階を経て導かれ、その段階を一つ経るごとにプログラムの量は何倍にもなります。これに、計算の並列化のためのプログラムや、高性能を発揮するための変換がさらに加わることで、スーパーコンピュータでのシミュレーションに必要なプログラムは、ときに数十万行にも及びます。このようなプログラムの作成やチューニングは大変困難です。

一方で、原理的にはシミュレーションしたい自然現象とその離散化法を指定すれば、プログラムは機械的に生成できます。このようなツールの実現は、長年待ち望まれています。しかし、プログラミングはシミュレーションをしたい対象と、プログラムを実行するコンピュータの双方に対する深い理解を前提として成り立つ複雑な作業です。方程式の記述から、スーパーコンピュータの性

能をも引き出せるようなプログラムを汎用的に生成することは、これまで誰もできませんでした。

2. 研究手法と成果

コンピュータはもともと、計算命令を一つずつ（逐次的に）実行するものとして誕生し、現在普及しているプログラミング言語も命令の逐次実行が基本思想にあります。しかし、スーパーコンピュータのプログラムではこの基本思想が適しているとはいえません。この思想に従いプログラムを作成するには、スーパーコンピュータを構成する一つの計算機が担当する、ごく一部に注目して、逐次実行されるプログラムに翻訳しつつ、コンピュータ全体でタイミングを合わせて動作させる、という手順を踏む必要があります。一糸乱れぬマスをゲームを、一人一人に詳細な指示を与えて完成させるような、膨大な手間がかかっていました。

共同研究グループは、自然が元来備えている「局在性」と「並列性」を、なるべく分解せずにスーパーコンピュータ（並列計算機）の動作に変換できればよいと考えました。そこで、もともとの方程式があらわす自然現象を一気にシミュレーションできるスーパーコンピュータがあると仮定し、方程式があらわす対象をその仮想スーパーコンピュータのプログラムに翻訳する、という方式を考案しました。この仮想スーパーコンピュータのプログラムを、現実のスーパーコンピュータが持つ階層構造にあてはめながら実際のプログラムに翻訳する手法を検討した結果、方程式からプログラムまでの変換を自動的に実現できる一連の段階を発見しました。そして、すべての段階に対応する数学的定義を作ることで、新たなプログラミング言語「Formura」を開発しました。これによって、これまではシミュレーションとコンピュータ双方に深い知識が必要であったスーパーコンピュータのプログラミングを、機械化することに成功しました。さらに Formura は、同じアプリケーションに対して何万通りものプログラムを生成し、「京」の高性能を活かしてそれら候補をすべて実際に実行して性能を測定し、最も速かったものを自動的に選択します。

また、共同研究グループは、実際に Formura をもちいて記述したプログラムで、地下に伸びる菌類の複雑な挙動をシミュレーションすることに成功しました（図 1）。このとき Formura は、3 万行以上ものプログラムを 4 万通り以上、自動的に生成しました。一方で、その元となったシミュレーションプログラムはわずか 20 行あまりで、はがき一枚に収まる程度でした。

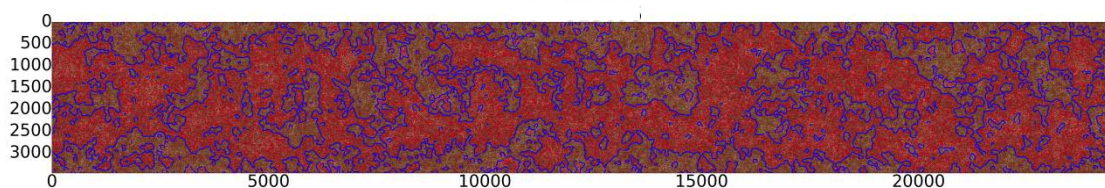


図 1 Formura で作成したプログラムによってシミュレートされた地下の生態ネットワーク捕食者（緑）が被捕食者（赤）に対して優勢な領域が大規模なクラスタ（青線）を作ることが示された。

縦軸・横軸の単位はミリメートル。先行研究である Pearson らが 1993 年に 2 次元で行ったシミュレーションを 3 次元に拡張し、同じ振る舞いがおこることを再現して確かめた。解像度が大幅に上がったことで、大規模な捕食者クラスタの成長も観測できている。

3. 今後の期待

開発したプログラミング言語「Formura」により、規則格子シミュレーション分野においては、自然科学者が慣れ親しんだ方程式の記法を使ってシミュレーションしたい対象を記述することで、「京」などの多数の計算機を協調して動作させるスーパーコンピュータの性能を引き出せる高度なプログラムが、自動かつ汎用的に生成可能となりました。気象、地震、宇宙、生態ネットワークの研究など規則格子シミュレーションを用いる分野において、シミュレーション研究の加速が期待できます。

また、Formura を開発したことで、規則格子以外のシミュレーション分野においても、アプリケーションや情報科学からの研究者の参入を促し、プログラム自動生成技術の実現に向けた研究の活性化が期待できます。

4. 論文情報

<タイトル>

Automatic generation of efficient codes from mathematical descriptions of stencil computation

<著者名>

Takayuki Muranushi, Seiya Nishizawa, Hirofumi Tomita, Keigo Nitadori, Masaki Iwasawa, Yutaka Maruyama, Hisashi Yashiro, Yoshifumi Nakamura, Hideyuki Hotta, Junichiro Makino, Natsuki Hosono, Hikaru Inoue

<雑誌>

FHPC 2016: Proceedings of the 5th International Workshop on Functional High-Performance Computing

<DOI>

10.1145/2975991.2975994

<タイトル>

Simulations of below-ground dynamics of fungi: 1.184 pflops attained by automated generation and autotuning of temporal blocking codes

<著者名>

Takayuki Muranushi, Seiya Nishizawa, Hirofumi Tomita, Keigo Nitadori, Masaki Iwasawa, Yutaka Maruyama, Hisashi Yashiro, Yoshifumi Nakamura, Hideyuki Hotta, Junichiro Makino, Natsuki Hosono, Hikaru Inoue

<雑誌>

SC '16 Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis Article No. 3

5. 補足説明

[1] スーパーコンピュータ「京（けい）」

文部科学省が推進する「革新的ハイパフォーマンス・コンピューティング・インフラ(HPCI)の構築」プログラムの中核システムとして、理研と富士通が共同で開発を行い、2012年に共用を開始した計算速度10ペタフロップス級のスーパーコンピュータ。

[2] 離散化法

もともと連続的である自然現象を、有限個の要素で近似的に表す方法。離散化（数値化）により、コンピュータシミュレーションが可能になる。

[3] 自然が元来備えている「並列性」と「局所性」

すべて自然現象には、光速・音速・菌糸の伸びる速さなど、伝達速度に限りがある。このため、一か所で起こったことの影響は、最初はその近傍にしか影響を及ぼさないという「局所性」があり、また無数の構成要素が、お互いに影響を及ぼしあいながらも、それぞれが同時並行して発展していくという「並列性」がある。

[4] 規則格子シミュレーション

シミュレーションの離散化法のうち、離散化対象を縦横の規則正しい格子に区切って行うものこと。

[5] ゴードン・ベル賞

ゴードン・ベル賞(ACM Gordon Bell Prize)は、並列計算技術の向上を目的にACM(Association for Computing Machinery、米国計算機学会)によって運営され、毎年11月に開催されるハイ・パフォーマンス・コンピューティング（高性能計算技術）に関する国際会議（International Conference for High Performance Computing, Networking, Storage and Analysis）で、ハードウェアとアプリケーションの開発において最も優れた成果を上げた論文に付与される賞。