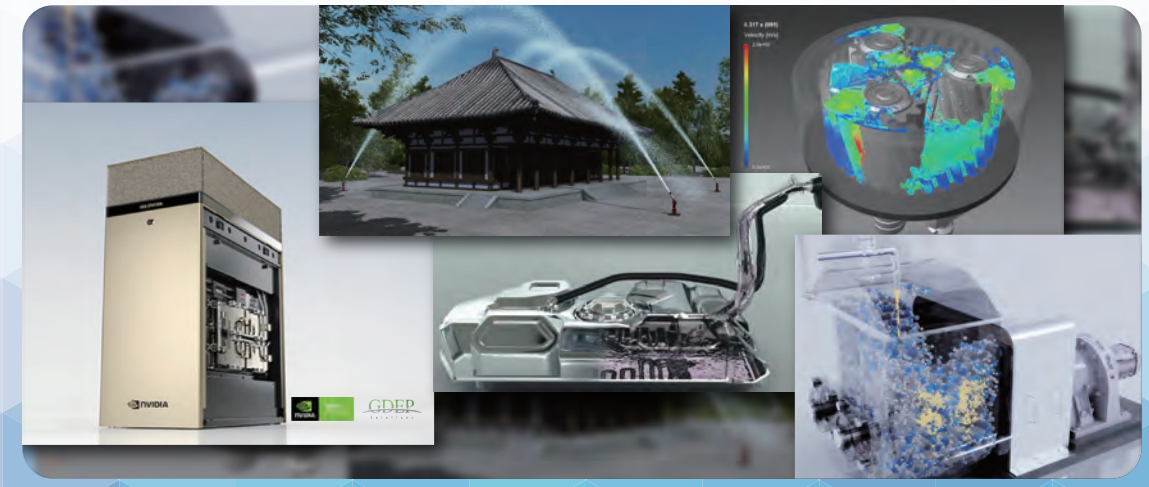
PROMETECH.

ソリューションガイド



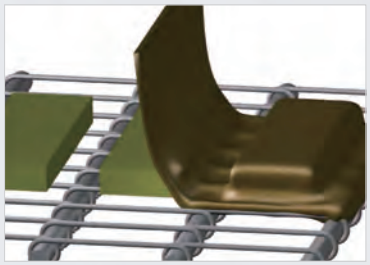
粒子法流体解析ソフトウェア

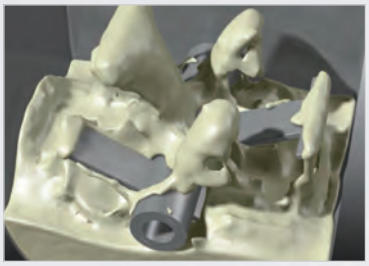
Particleworks®
Particle-based simulation software for CAE

Particleworksは、水や油などの液体挙動を評価するためのメッシュレス流体解析ソフトウェアです。トランスミッションやエンジンの潤滑、モーター冷却、冠水路走行、薬品や樹脂、食品などの撹拌・混練から、土砂災害・洪水まで幅広い分野の流体問題をシミュレーションします。

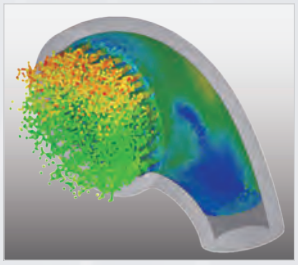
適用例

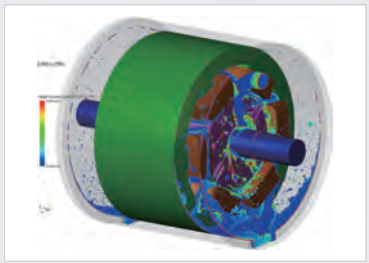
※一部CGレンダリング加工しています。


チョコレートのコーティング

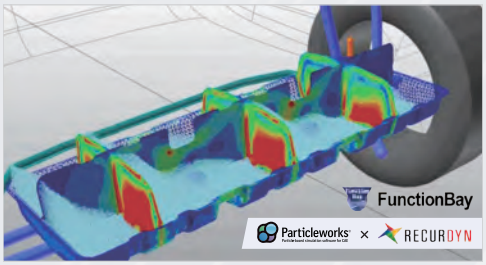

高粘性流体の撹拌解析

洗浄層の撹拌解析

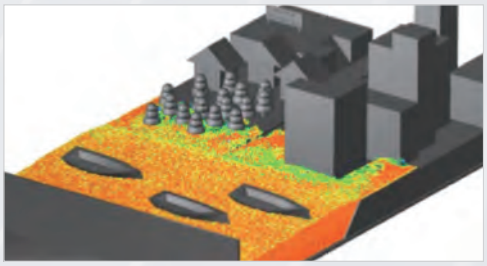

シャワーヘッド内の水流解析


モーターの冷却用オイルの挙動解析

モデルデータ：JMAGアプリケーションカタログNo.018
https://www.jmag-international.com/jp/catalog



走行中のオイルタンク内圧力によるバッフルの変形と応力事例 ＜流体-弾性体連成＞

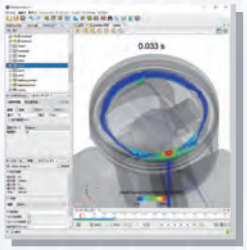

津波遡上解析

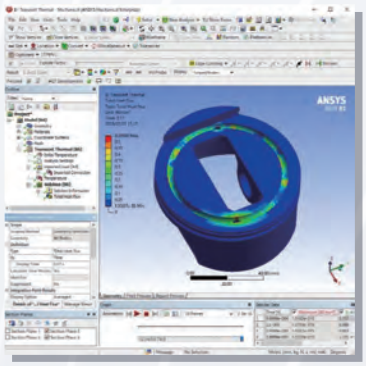
Particleworks-Ansysインターフェイス

Particleworks for Ansys

ParticleworksをAnsys Workbench環境に統合し、Particleworksの結果（圧力と熱伝達係数）をAnsys Mechanicalの境界条件として自動マッピングします。複雑な液体挙動を考慮し、より正確な構造計算が可能になります。

Particleworks® for Ansys


Particleworksによる
ピストンオイル挙動解析

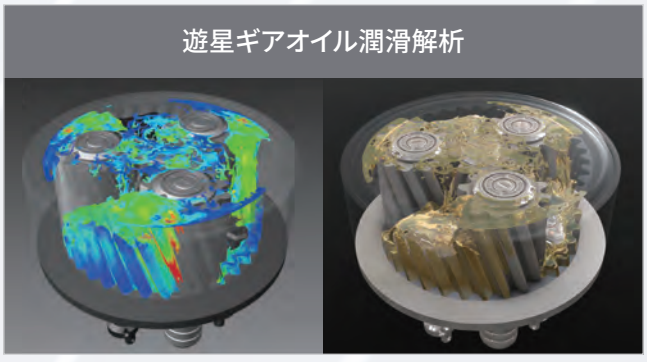


Ansys Mechanical に
熱伝達係数を荷重としてインポート

可視化オプション

Visualization Option

Particleworks/Granuleworksの解析結果を汎用CGレンダリング・編集ソフトウェアへ容易に移行し、ビジュアライゼーションにかかる時間を大幅に短縮するライセンスオプションです。CAE解析結果からの動画作成、XR（VR/AR/MR）コンテンツ開発を促進します。



Particleworksビュー　Unityによるレンダリング

DEM粉体解析ソフトウェア

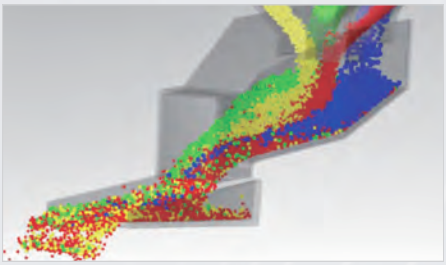
Granuleworks®

Advanced Simulator for Granular Materials

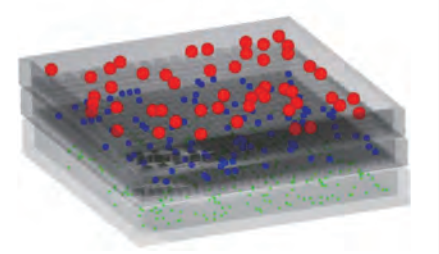
Granuleworksは、離散要素法(DEM)の理論に基づく粉体の解析ソフトウェアです。食品、医薬品、化学品、輸送機器、電子材料など、粉体を使った様々な製造プロセスや粉体加工、粉体装置の設計・改良に活用することができ、混合、搬送、充填など粉体の現象を容易に解析できます。

適用例

リボンミキサーによる攪拌解析

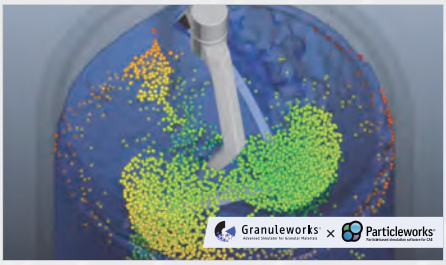


複数粉体の合流



振動分級解析

円筒容器内の粉体混合解析
提供:月島機械株式会社様



液体-粉体連成解析
撹拌槽挙動解析

機構-粉体連成解析

PROMETECH.

Visualizationサービス

プロメテック・ソフトウェアでは、最新のComputer Graphics技術を用いたVisualizationサービスをご提供します。解析結果やCADデータ、図面、写真資料などをもとに、簡易的なビューワでは難しい特殊表現や高品質でフォトリアルなCG 映像を作成します。動画だけではなく、インタラクション可能なXR(VR/AR/MR)コンテンツまで幅広く対応します。

お好きな時間・場所でご覧いただける

Webセミナー開催中

オンデマンド動画ストリーミング配信で、プロメテック・ソフトウェアの製品・サービスを紹介する無料のWebセミナーを開催しています。視聴お申込後は、ご都合の良いタイミングでご覧いただけます。ぜひご参加ください。

粒子法流体解析ソフトウェア Particleworks
【概要紹介編】【解析事例編】【ソフトウェア操作編 ～複合流体撹拌解析～】

粉体解析ソフトウェア Granuleworks
【概要紹介編】【解析事例編】【ソフトウェア操作編 ～振動ふるいの解析～】

Visualizationサービス / Visualization Option ご紹介編

Particleworks for Ansys ご紹介編

Particleworks トピックスセミナー
ゴム混練機への適用／ギアボックスのオイル撹拌／エアコン室外機の降雨シミュレーション／フレッシュコンクリートのL型フロー試験／自動車の空調防水性の設計及び性能評価

セミナー情報の詳細はこちら

https://www.prometech.co.jp/web_seminar_outline_ja.html

HPCコンサルティングサービス

長年HPCシステムにおける最適化を研究してきたエンジニアが携わり、お客様のHPC技術の課題に最適なソリューションを提案いたします。

提案サービス

1. プログラムの性能解析による最適化・高速化支援
2. プログラムの性能特性や利用方法に応じた最適な並列化手法・開発方法の提案・相談
3. プログラムのマルチスレッド対応やGPU対応などのコード移植作業
4. ご利用のソフトウェアやアプリケーション、予算などに合わせた最適なコンピューターシステムの提案・相談
5. その他HPCに関する技術支援等

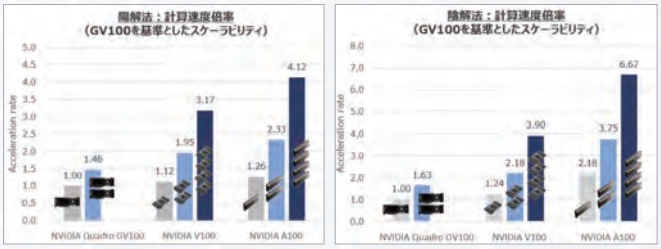
コミュニティサイト「HPC WORLD」

ハードウェアやシステムの導入サポート

プロメテック・グループには、AIやCAD/CAE分野を中心にNVIDIA社のGPU製品を始めとした各種ソリューションを提供している「GDEPソリューションズ株式会社」があります。

マルチGPU環境で、驚愕のパフォーマンスを発揮するようになったParticleworksの新バージョンに最適なGPUおよびGPU搭載マシンの販売・レンタルを行なう他、ストレージやネットワーク、更には各種ライブラリ・コンパイラ・開発ツールの提供も行なっています。

ハードウェアやシステム導入のご相談は、技術コラムも満載のGDEPソリューションズWebサイトからお気軽にお問合せください。



マルチGPU環境でパフォーマンスがスケールするParticleworks

Particleworks おすすめ
「NVIDIA® Quadro GV100」「NVIDIA V100S Tensor Core」「NVIDIA A100 Tensor Core」

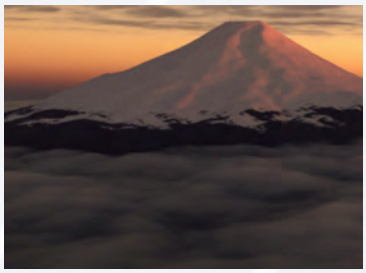
最適なマシンのご提案、GPU搭載作業～動作確認、導入まで、
安心してすぐにお使いいただける環境をトータルにサポートしています。

GDEP Solutions　URL：www.gdep-sol.co.jp

プロメテックCGリサーチ

プロメテックCGリサーチは、プロメテック・ソフトウェア(株)内にあるCG分野の研究グループで、全く新しいCG技術を研究・開発します。次世代のパイオニアを排出することを目標として、CG分野の未解決の問題に対して研究活動を行います。主に、学術的な研究活動を行い、学会発表、論文を通して技術発信をします。

なお、より一層のCG研究を進めていくため、共同研究先を募集しています。共同研究をご希望される際はcgr-kyodo@prometech.co.jpまでご連絡ください。



PROMETECH.
プロメテック・ソフトウェア株式会社

【本　社】
〒113-0033 東京都文京区本郷三丁目34番3号 本郷第一ビル8階
Tel: 03-5842-4082 Fax: 03-5842-4123

【名古屋オフィス】
〒460-0003 愛知県名古屋市中区錦一丁目17番26号 ラウンドテラス伏見3階
Tel: 052-211-3900 Fax: 052-211-3901

https://www.prometech.co.jp/
sales@prometech.co.jp

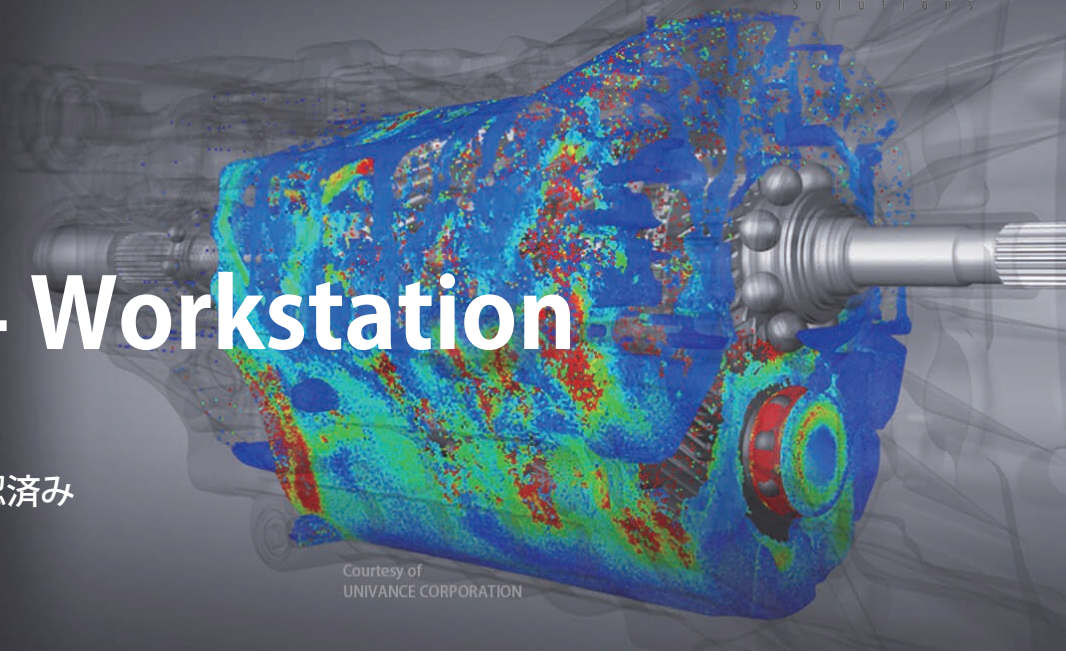
NVIDIA ELITE PARTNER

GDEP Solutions

NVIDIA

GPU 最大2基 搭載

HP Z8 G4 Workstation



解析ソフトウェア 動作確認済み
推奨ワークステーション

搭載おすすめGPU

NVIDIA Quadro GV100

NVIDIA® Quadro® GV100は、倍精度演算性能が 7.4 TFLOPS と高く、GPUメモリは大容量の32GBを搭載しています。高精度、大規模なシミュレーションを安定して解くことができます。

NVIDIA GPU 世代別 スペック

2013年 → 現在

	Tesla K40	Quadro GP100	Quadro GV100
CUDAコア数	2880	3584	5120
GPUメモリ	12 GB GDDR5	16 GB HBM2	32 GB HBM2
メモリバンド幅	288 GB/s	732 GB/s	870 GB/s
倍精度演算性能	1.43 TFLOPS	5.13 TFLOPS	7.4 TFLOPS
単精度演算性能	4.29 TFLOPS	10.25 TFLOPS	14.8 TFLOPS

おすすめワークステーション

HP Z8 G4 Workstation

HP製品の安定した品質・スピーディーな保守対応により、弊社一押し製品!

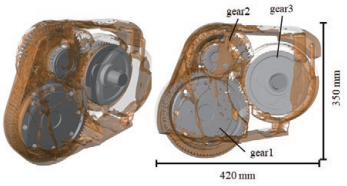
✓ GPU は最大2基搭載
1基搭載から始めて、さらに大規模な解析や、計算時間の短縮を求めて、後付けでGPUを追加することも可能

✓ インテル® Xeon® Platinum 8100 プロセッサー ファミリーに対応し、デュアル・プロセッサー構成時で最高48コア・96スレッドのシステムを実現

✓ 24本のメモリスロットを装備し、最大システムメモリ容量も1.5TB まで搭載できるハイエンドモデル

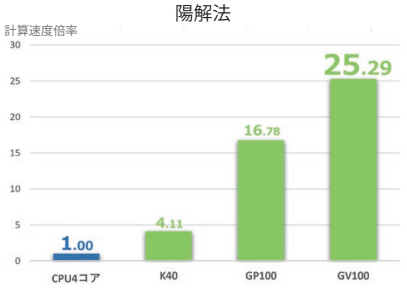
計算速度比較

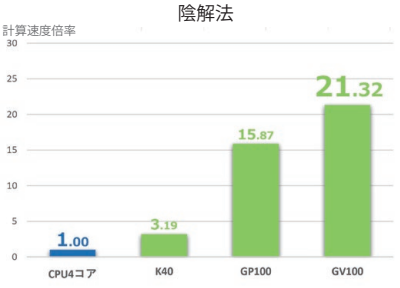
CPU 4コア vs GPU 1基



CPU 4コアに対して、
NVIDIA Quadro GV100 1基の計算速度は、
陽解法で 約25倍の高速化

時間で例えると・・・
25時間かかっていた計算時間が、たったの1時間に!





粒子法 流体解析ソフトウェア Particleworks Ver 6.2.0
粒子数：陽解法＆陰解法 460万規模
OS：Windows 10
CPU：インテル® Core™ i7-6805K プロセッサー

GPU：NVIDIA® Tesla® K40
NVIDIA® Quadro® GP100
NVIDIA Quadro GV100

解析ソフトウェア 動作確認済み

HP Z8 G4 Workstation

NVIDIA Quadro GV100 搭載モデル

ご要望に応じた仕様変更もできます。
GPUを4基搭載したい方 ・ GPUの種類を変更したい方 ・ サーバータイプを検討したい方 など、業務に適した製品・スペックをご提案いたします。 お気軽にお問い合わせください。

GPU搭載数	1基	2基
GPUメモリ	32GB(Quadro GV100× 1)	64GB(Quadro GV100×2)
CPU	インテル® Xeon® Silver 4108 プロセッサー (1.8GHz, 8コア, 11MB, 2400MHz) × 1	インテル® Xeon® Silver 4108 プロセッサー (1.8GHz, 8コア, 11MB, 2400MHz) × 2
システムメモリ	96GB DDR4 SDRAM (2933MHz, ECC, Registered) シングルプロセッサー用	
SSD	256GB HP Z Turboドライブ G2 (内蔵M.2スロット接続 TLC SSD)	
HDD	2TB ハードディスクドライブ (SATA, 7,200rpm)	
描画用GPU	NVIDIA Quadro P400, 2GB	
ディスプレイコネクタ	DisplayPort ×3 , HDMI ×1 , USB Type-C ×1 *DisplayPortは、GPUにより異なります。	
OS	Windows 10 Pro for Workstations 64bit 日本語版 or Ubuntu	
光学ドライブ	DVD+/-RW Slimline	
保守	3年間翌日オンサイト（休日修理付き）* 障害時HDD返却不要 * GPUはセンドバック保証3年間	
筐体サイズ	幅 216mm × 奥行 556mm × 高さ 445mm	

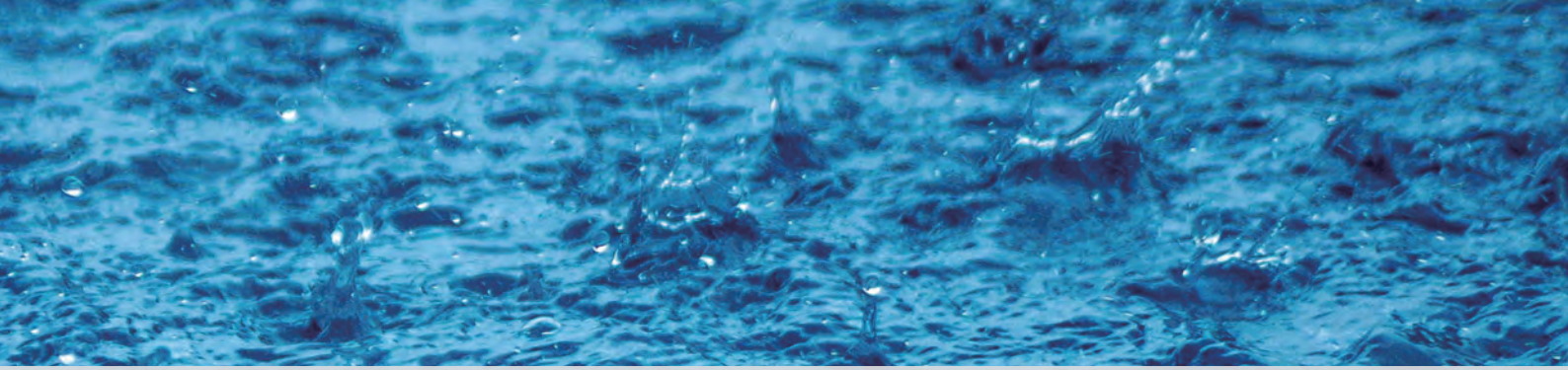
詳細や参考販売価格については、下記のWEBページよりご確認ください。

https://www.gdep-sol.co.jp/hp-z8-g4-cae

GDEP ソリューションズは「NVIDIA エリート・パートナー」に認定されています。
NVIDIA DGX、GPU のプロフェッショナルとして、AI・データサイエンス・数値計算・解析など、用途に最適な製品のご提案、導入までトータルにサポートしています。また、セキュリティなどのクラウド製品も取り扱っています。
あらゆるシーンでお客様の追い求める様々な挑戦をサポートします。

●お問合せは

GDEPソリューションズ株式会社
東京都文京区本郷三丁目34番3号 本郷第一ビル8階
TEL：03-5802-7050 FAX：03-5842-4123
E-mail：gdepsales@gdep-sol.co.jp
https://www.gdep-sol.co.jp



PROMETECH. GPUコンサルティングご利用ユーザーインタビュー

山口大学 大学院創成科学研究科 様

乱流現象を解明するためのシミュレーションをGPUコード化し計算速度を飛躍的に向上

山口大学 大学院創成科学研究科 准教授の鳴海孝之様は、工学部の学生向けに物理学を体系的に理解するための工学基礎教育を行う傍ら、統計物理学の考え方を背景に、理論解析や計算機シミュレーションによる様々な非線形現象を研究されています。これまでに、ミツバチの造巣、液晶電気対流、過冷却液体状態など幅広い分野を対象とされていますが、今回GPU化を進めた鳴海先生が書かれたシミュレーションプログラムは、乱流を抽象化した数理モデルを理論的に研究するためのものです。

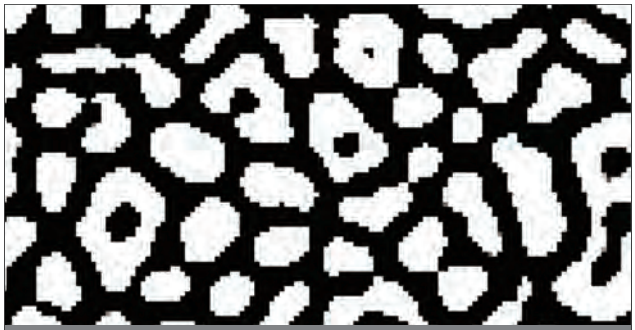
山口大学 大学院創成科学研究科 准教授
鳴海 孝之 様

GPUコード化課題背景

- 偏微分方程式によるプログラムの計算量増大に対応したい。
- GPUを使って、劇的に計算環境を改善したい。
- ブラックボックス化せず、継続的にプログラムを書き換えたい。

ご研究のプログラムをGPU化された背景をお聞かせ下さい。

（鳴海先生）乱流の研究は多様で、例えば完全に乱れ切った乱流の研究は比較的多いのですが、一方で流体がどのように乱流になるのかは、あまり明らかになっていません。私が着目しているのは、全体を平らにする効果、つまり水面に波が立たないようにする効果と、波が出来た後に波の面が凹凸になる効果の両方が入っている現象で、それが競合して乱れが生じる際にどのような性質が現れるかを、数理モデルによるシミュレーションを用いて研究しています。以前は1つのCPUで並列化もせずに計算していました。ですが、研究が進むに連れて計算時間が大分かかるようになり、限界を感じ始めていたのです。その数理モデルというのは、偏微分方程式を解くものなのですが、系全体を非常に細かくメッシュを切って計算しますので、GPUにすることのメリットが大きいのです。GPU化はかねてから実現したいと思っていましたし、同じ手間をかけるなら、思い切って手を入れたいと考えました。OpenMPなども候補としてありましたが、チューニングにかける時間や実際に得られるメリットを考えると、期待する効果が得られないという経験もありましたし、GPUなら劇的に改善する可能性があると判断したのでお問い合わせいたしました。もともとプロメテック・ソフトウェアのグループ会社であるGDEPソリューションズからGPUマシンを購入していたので、そこから情報を辿ってGPU化のサービスを見つけたというのが経緯です。



数値計算で得られる波の振幅を可視化

GPU化はどのような流れで行いましたか。

実際に動いたのが2021年に入ってからですが、大学は4月から6月が前期で忙しいので、その期間に高速化していただいて、時間の取れる夏休みにコードが手元にあると嬉しいと思いまして、そのようなスケジュールでお願いしました。

（担当者）1月下旬にお問い合わせいただき、当初からGPU化をご希望だったことと、お伺いしたプログラムの内容からGPUでの高速化が可能だと判断できたので、コンパイルを担当させていただきました。最終的に行いたい二次元問題に対応させる前に、まず一次元問題のプログラムをお預かりし、それでGPU化が出来そうかを確かめました。その後、結果をどう

合わせるか、二次元だとどうなるかなどを話し合いながら対応を進めました。今回のプログラムがC++のオブジェクト指向型のモデルで、GPUを実装するためのOpenACCという言語と若干相性が良くないため、そこを解決するのに若干の時間を要しました。結果的には、約2ヵ月のやり取りとりとコード化作業で、期限の6月末までに完成し納品に至ったという流れです。

私は、とにかく早く対応して下さったという印象を持っています。問い合わせのメールにもすぐに返信して下さいましたし、サービス対応に関しては大変満足しています。それから、最初の打ち合わせの段階から重視していたのは、プログラムの可読性つまりブラックボックスにして欲しくない、ということです。テクニックを駆使してすごいコードに書き換えても、私自身が全く理解できないコードになってしまうと、いくら今回高速化が出来たとしても、今後誰も手を加えられずこれ以上開発できなくなってしまいます。その意図をきちんと汲み取ったうえで高速化して下さったのが非常に助かった部分です。

私たちもそこは重視しており、コードが進化していく時に誰でも読めて開発の持続性を損なわないようにしないと、せっかくGPU化したものも無駄になってしまいます。HPCを見ている情報工学側の人間ではなく、実際に計算をされる計算科学の方々が読めて使いやすいものにするということを心がけました。そのメンテナンス性の良さというのは、非常に大事だと考えています。

実際GPUで、高速化や使い易さはどう変わりましたか。

高速化については現在どれくらいの差が出ているかを検証していますが、もちろん確実に速くなっています。比較するとおおよそ5～6倍になっています。特に評価できるのが、計算格子や計算領域が増えてシステムのサイズが大きくなっても、その性能がかなり担保出来ており、4～5倍の高速化を保てています。

GPUはどちらかというと、系が大きく計算領域が大きい方が得意なハードウェアなので、大きければ大きいだけ、その優位な差が見えてきます。今回のテストは比較的小さめの系で行いましたが、それでも差はしっかりと出ていましたので、大きくすればさらに差が広がっていきます。その意味では今回のコードは計算量が大きくGPUにとても適しています。

何がGPUに向いているかは、どのように判断するのでしょうか。

お客様の話をお伺いし、フーリエ変換なのか、行列積なのか、あるいは陰解法と呼ばれる連立一次方程式を解くコードなのか、などのようなパターンで傾向が決まってくると思います。その時にGPUが適しているのか、CPUの方がもしかしたらいいかもしれないとか、多少のヒントになります。プログラムをコンパイルするには、お客様と私たち開発者側の連携や情報共有がとても重要なポイントになります。ですから、最初に鳴海先生から作業に取り掛かるのに十分な情報をいただけましたので、非常にやりやすかったです。

その情報共有というところでは、私もかなり好印象で、最初の打ち合わせの時から本当に正確なアドバイスをいただき、こちらも非常に助かりました。GPUとも相性があって何でもGPUで上手く出来るわけではないことも説明下さいました。こちらはまだサービスを利用するか未定の状態で全部はお話できない中で、丁寧に聞いて下さり私もある程度の感触がわかったことが、発注する上で大きな判断材料になりました。コロナ下ということで、実はまだ直接お会いしておらず、最初からメールとオンライン会議だけで打ち合わせを続けておりますが、全く問題はないですしサポートにも非常に満足しています。

今後の研究で取り組みたいことや、将来的なビジョンなどをお聞かせ下さい。

昔からある身近な乱流という分野で、多くの研究者がそれぞれ違った視点で研究をされている中で、私は今続けている自分の基礎研究で乱流現象を明らかにして、応用技術としても何か貢献できればと思っています。そのためには数値計算は必要不可欠ですので、プログラムの技術を磨いて、効率よく計算してできるだけ結果を出していきたいです。プロメテックさんには、本当にこちらの要望を納期やコードの編集も含め、全て聞いていただいた形になっていると思います。ですので、本当に感謝しかないですし、今後もぜひ機会があればサービスを利用したいと思っています。

本記事の内容は、取材時2021年8月の情報です。製品の機能および構成などは取材時より変更されている可能性がありますので、予めご了承下さい。最新の情報については、プロメテック・ソフトウェア(sales@prometech.co.jp)までお問合せください。

お客様紹介

山口大学

長州藩士「上田鳳陽」によって、1815年に創設された私塾「山口講堂」を起源とし、明治・大正期の学制を経て、1949年に、地域における高等教育および学問研究の中核たる新制大学として創設されました。現在は、9学部8研究科からなる学生数1万人を超える総合大学となっています。
ホームページ： http://www.yamaguchi-u.ac.jp

取材日：2021年8月30日

今回のGPUコード化の目標

今回のケースでは、鳴海先生ご自身でプログラムを開発されている経緯から、GPUを用いた計算の高速化をメインの目的にせず、今後、先生がメンテナンスしやすい状態でGPUに対応させる、という点を重視し以下の目標を設定しました。

1. コードの基本的な構造は変えずに実装すること
2. GPUで効率が悪い場合は限定的にコードを最適化すること
3. もちろんGPUによりオリジナルコード（逐次処理）よりも高速化されること

実際のOpenACC化作業

上記の目標を達成するため、以下の作業を主に実施しました。この前段階として技術的調査も行っています。

1. std::vector<T>クラステンプレートをthrust::universal_vector<T>クラステンプレートへ置換
2. FFTWライブラリをcuFFTライブラリへ書き換え
3. 主要な計算関数をOpenACC化

std::vector<T>クラステンプレートの置換

前述のインタビューでもふれましたが、今回ご依頼いただいたプログラムはC++のオブジェクト指向プログラミング（OOP）の側面を活用しており、OpenACCにとって少し苦手とする内容になっています。特にメモリ管理が複雑になるのですが、この話だけで大きなトピックとなるため今回は割愛いたします。

C++の標準仕様であるstd::vector<T>クラステンプレートは連続なメモリ領域として配列を確保するクラステンプレートで、メモリの確保および解放がC++の機能を使って自動化されています。クラステンプレートとはTに任意の変数型を指定する機能で、まさにテンプレート（雛形）としてクラスを作成する手段です。Tにはint，floatといったスカラ型だけではなく、構造体やクラスを指定することも可能です。またアロケーターというvector<T>内でのメモリ管理を担当するクラスを自作し、今回のようにOpenACCで使用するオブジェクトのメモリ管理を自動化することも可能です。

今回はvector<T>と同等のインターフェイスを持つ、CUDA Thrustライブラリのthrust::unviersal_vector<T>クラステンプレート1）へvector<T>を置き換えました。thrust::universal_vector<T>はNVIDIA GPUのネイティブAPIであるCUDAのunified　memoryという機能を用い、データ転送も含めCPUとGPU間のメモリ管理をすべて自動化するためのクラステンプレートです。CUDA 11.4またはNVIDIA HPC SDK 21.7から対応したThrustライブラリがバンドルされています。

OpenACCではネイティブAPI（CUDA）との相互連携も提供されており、CUDAで確保したGPUメモリをOpenACCディレクティブ内で使用することが可能です。以下のコード例のように、GPUメモリを使用したいOpenACCディレクティブにdeviceptr(p)句を記述するだけです。

thrust::universal_vector<T>クラステンプレートにより、vector<T>でも得ていたメモリ管理の自動化に加え、CPU-GPU間のデータ転送の自動化も得られました。煩わしいメモリ管理は一旦忘れてOpenACC実装に集中することができます。

```
thrust::universal_vector<T> ux(nx);
auto p = thrust::raw_poitner_cast(ux.data());

#pragma acc kernels deviceptr(p)
for (int i = 0; i < nx; ++i)
  p[i] = i * 0.5;
```

図1：acc deviceptr(p)を用いたサンプル

FFTWライブラリをcuFFTライブラリへ書き換え

今回のプログラムは連立一次方程式を解くためにFFTWを使用していました。NVIDIA GPUの場合、FFTWとデータレイアウトが互換なCUDAで実装されたcuFFTライブラリが提供されています。cuFFTはFFTWのインターフェイスをそのまま使用してGPUでの高速化を図ることもできますが、この手法ではFFTWの計算ルーチンを呼び出す際に都度CPU-GPU間のデータ転送が行われて効率がかなり悪くなる場合があるため、FFTWの呼び出しをすべてcuFFTに書き換えました。

データレイアウトはFFTWと互換なので、cuFFTのインターフェイスに変換するだけです。非常にシンプルなため、ドキュメントにもAPIの対応表のみが記載されています 。以下にFFTWとcuFFTで同じ計算を行う場合のサンプルコードを示します。

```
FFTW
const int nx = ..., ny = ...;
const int nky = ny / 2 + 1;
fftw_complex *fftw_in  = (fftw_complex*)malloc(sizeof(fftw_complex)*nx*ny);
double       *fftw_out = (double*)malloc(sizeof(double)*nx*nky);
fftw_plan     plan
= fftw_plan_dft_c2r_2d(nx, ny, fftw_in, fftw_out, FFTW_PATIENT);
...
fftw_execute(plan);
```

```
cuFFT
const int nx = ..., ny = ...;
const int nky = ny / 2 + 1;
thrust::universal_vector<cufftDoubleComplex> cufft_in(nx*ny);
thrust::universal_vector<double>             cufft_out(nx*nky);
cufftHandle plan;
cufftPlan2d(&plan, nx, ny, CUFFT_Z2D);
...
cufftExecD2Z(plan, cufft_in, cufft_out);
cudaStreamSynchronize(0);
```

図2：FFTWとcuFFTの使用方法の違い

主要な計算関数をOpenACC化

メモリ管理とライブラリの入れ替えが完了すれば、あとはOpenACCディレクティブを追記し各計算をGPUで計算できるようにするだけです。

オリジナル実装では動的な2次元配列を確保し計算に使用していましたが、C言語の動的多次元配列の場合、メモリアクセスが不連続になってしまう可能性があります。そこで配列をすべて1次元に書き換え、データレイアウトとアクセス方法を工夫しメモリ上連続な2次元配列としました。

```
std::vector<std::vector<double>> v;
v.resize(nx, std::vector<double>(ny));
for (int ix = 0; ix < nx; ++ix)
  for (int iy = 0; iy < ny; ++iy)
    v[ix][iy] = ix * 2 + iy * 0.5;
```

```
std::vector<double> v(nx*ny);
for (int ix = 0; ix < nx; ++ix)
  for (int iy = 0; iy < ny; ++iy)
    v[ix * ny + iy] = ix * 2 + iy * 0.5;
```

図3：多次元配列の1次元配列への書き換え

GPU化の恩恵について

OpenACCとthrustライブラリを用いることで、メモリ管理の自動化とシンプルなGPU化を実現することができました。また、性能については評価用データについてNVIDIA V100 GPUを用いることでXeon W-2123 @3.6GHzに比べて3倍以上の性能を得られました。比較対象であるXeon　W-2123はワークステーション向けで単一コアの動作周波数が高いため、サーバーグレードのメニーコアXeonに比べて高い逐次処理性能を持ちます。これに対しV100　GPUの並列処理性能が3倍以上ですから、GPUがメンテナンス性を最優先した結果であることを考慮すると良好ではないかと思われます。

弊社では、今後ともお客様のご要望に合わせGPUのみならず様々なプロセッサやシステムに向けた最適化を提案してまいります。ご興味ありましたらご相談だけでもぜひご利用ください。

参考: 1）https://developer.nvidia.com/blog/support-for-cuda-unified-memory-now-available-in-thrust/

PROMETECH.
プロメテック・ソフトウェア株式会社
【本社】
〒113-0033 東京都文京区本郷三丁目34番3号 本郷第一ビル8階
Tel: 03-5842-4082　Fax: 03-5842-4123
URL: www.prometech.co.jp　E-mail: sales@prometech.co.jp