

第4回AVS可視化フォーラム 2019

並列・高速化を実現するための 高速化サービスの概要と事例紹介

株式会社アーク情報システム

営業部 仮野 亮

ソリューション技術部 佐々木 竜一

2019.08.30

はじめに

- アーク情報システムの紹介
- 高速化サービスとは？
- 事例紹介
- コンサルティングサービスについて

アーク情報システムの紹介

- 設立 : 1987年10月
- 資本金 : 3億600万円
- 従業員数 : 129名(2019年8月現在)
- 事業所 : 本社 市ヶ谷
横浜

アーク情報システムの紹介

■ 主な業務内容

- 科学技術計算（流体、構造解析。各種解析コードを用いた解析・開発 等）
- エンジニアリング系ソフトウェア開発（CAD、PDM 等）
- AI、IoT等 先端技術活用支援、開発
- 業務系システム基盤構築
- 映像・画像処理、AR・VR開発
- 高速化・並列化支援

高速化サービス

■ 高速化サービスとは？

- お客様のプログラムの高速化を調査・提案から実施までサポート
- 主なサポート言語
 - ◆ CUDA(Fortran, C/C++, Python)
 - ◆ OpenACC
 - ◆ MPI
 - ◆ OpenMP

高速化サービス

■ このようなお客様におススメ！

1. 自社解析プログラムの高速化効果が知りたい
2. 自社解析プログラムの実行時間に不満がある
3. 自社解析プログラムの並列化アドバイスがもらいたい
4. 自社で並列高速化を実施したいが方法が分からない
5. 高速化によりコストを削減したい

高速化サービス

■ 自社解析プログラム的高速化効果が知りたい

【ターゲット】

とにかくどのくらい速くなるのか知りたい

【アプローチ】

- ▶ 初期解析から高速化の可能性を診断
- ▶ 一部実装や経験値をもとに高速化効果を予測

※調査のみでも可能（具体的な実装はお客様で実施することでコスト削減）

高速化サービス

■ 自社解析プログラムの実行時間に不満がある

【ターゲット】

自社で並列高速化を試してみたが期待していた速度が出ない

【アプローチ】

- 高速化に最適なアルゴリズムを提案
- 高速化を阻害している要因を調査し提案

高速化サービス

■ 自社解析プログラムの並列化アドバイスがもらいたい

【ターゲット】

自社プログラムのどの部分をどう並列化すればよいか知りたい

【アプローチ】

- 並列高速化導入のサポート
- 高速化に有効なポイントを的確に提案

高速化サービス

■ 自社で並列高速化を実施したいが方法が分からない

【ターゲット】

自社で並列高速化を始めたいがどうすればよいか

【アプローチ】

- 並列アルゴリズムの学習サポート
- 並列高速化導入から実装までをコンサルティング
- プログラムに対して直接アドバイス

高速化サービス

■ 高速化によりコストを削減したい

【ターゲット】

解析計算の時間を短縮してコスト削減を実現したい

【アプローチ】

- 並列高速化を実施して時間短縮や作業効率をアップ
- 使用時間を短縮し、省電力化を目指す

高速化サービス

流れ

【Phase1】プロファイリング



1. お問い合わせ 2. NDA の締結 3. ソースのご提供 4. 初期解析 5. 作業方針検討

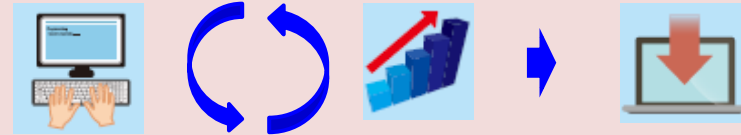
プロファイリングを行い、ホットスポットを見つけ高速化を検討

【お客様に提供するもの】
初期解析結果報告書
高速化提案書
⇒ 高速化の方針を決定

【ご用意頂くもの】

- (1) ソース一式
- (2) 入力データ (実行用)
- (3) 出力データ (結果確認用)

【Phase2】実装



6. 実装作業

7. 性能評価

8. 納品

【作業】

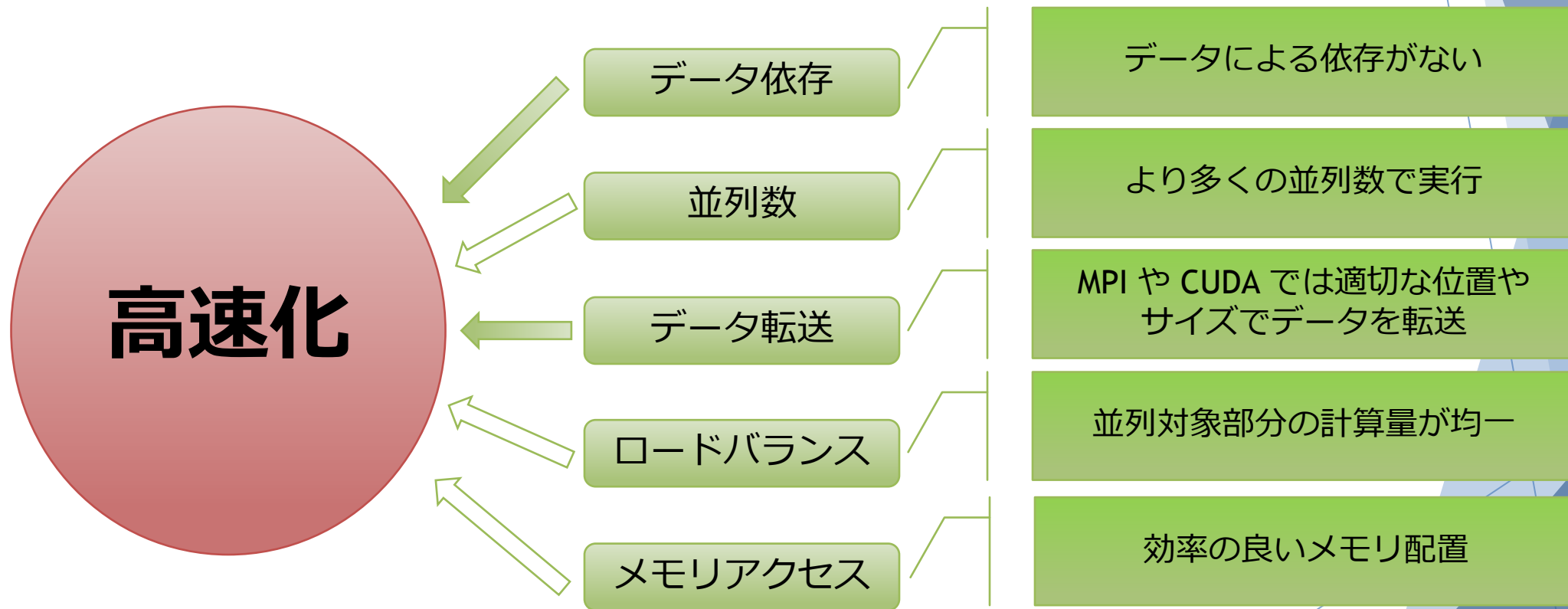
・実装および最適化
・ベンチマークによる性能評価
⇒ 繰返ししながら高速化を目指す

【納品物件】

・高速化結果
・報告書

並列高速化のポイント

- 大きな効果を得るために、並列化に適したアルゴリズムであることが重要



並列高速化のポイント

■ データ依存

データによる依存はないか？

- ▶ 並列処理を行う上で、依存がないことは大前提
- ▶ データ依存がある場合は？
 - ▶ データ依存性を無くすように再構築が必要

【事例：データ依存があるケース】

ループA内の依存がある処理を分割して別々のループで計算するように再構築する

```
do A loop
...
end do
```



```
do A loop
...
end do

do B loop
...
end do
```

並列高速化のポイント

並列数

より多くの並列数で実行

- GPUを使用する場合、コアをどれだけ遊びがなく稼働できるか？
 - 十分な並列数の確保（数万以上が望ましい）
 - 並列数を確保出来ない場合は？
 - ループ統合等の工夫が必要

【事例：並列数が少ないケース】

外側のループを分割し、内側のループと結合して大きなループを構築する

```
do middle loop
...
do small1 loop
...
end do

do small2 loop
...
end do
end do
```



```
do middle x small1 loop
...
end do

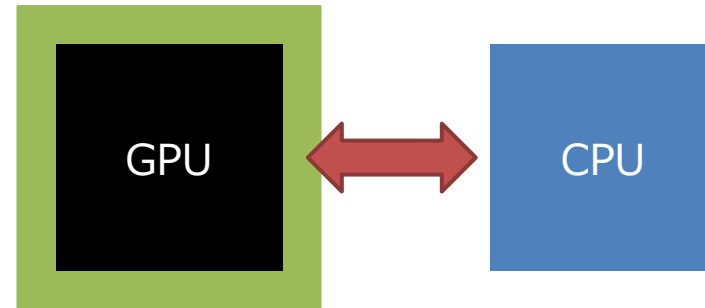
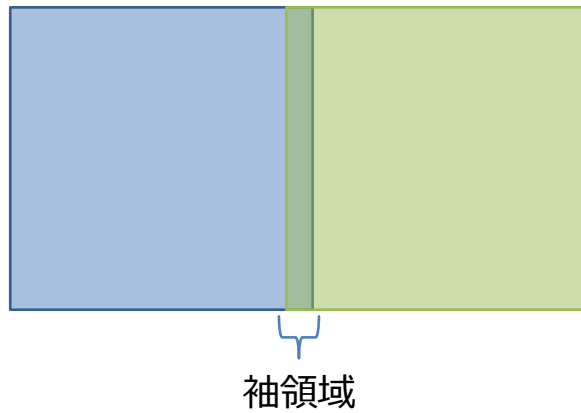
do middle x small2 loop
...
end do
```

並列高速化のポイント

データ転送

MPI や CUDA では適切な位置やサイズでデータを転送

- ▶ 転送に必要なデータの洗い出し
- ▶ ボトルネックになりやすい為、必要なデータのみ転送

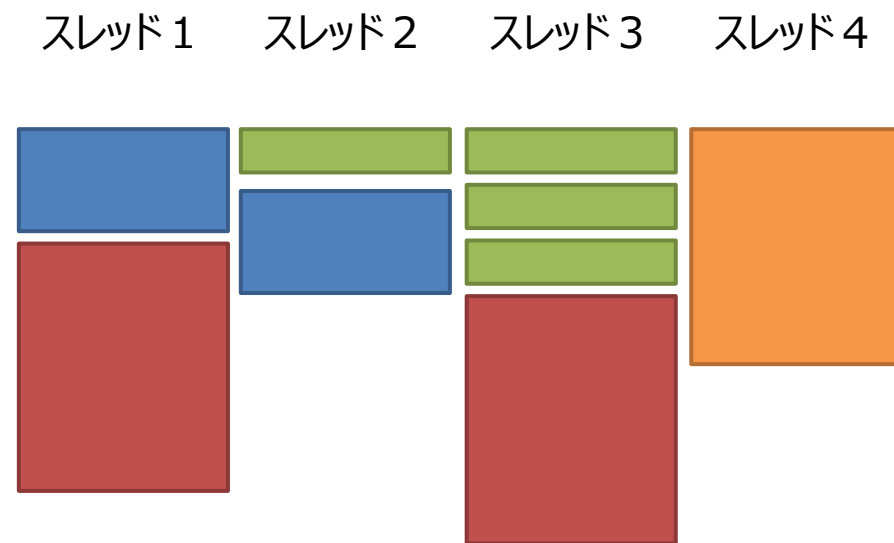


並列高速化のポイント

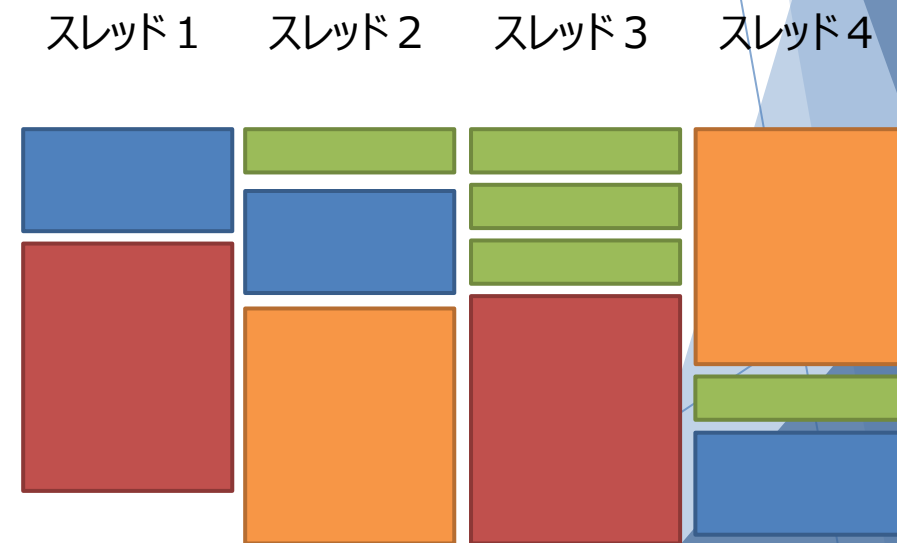
ロードバランス

並列対象部分の計算量が均一であるか？

- ▶ ロードバランスが悪いと高速化に弊害



ロードバランスが悪い例



ロードバランスが良い例

並列高速化のポイント

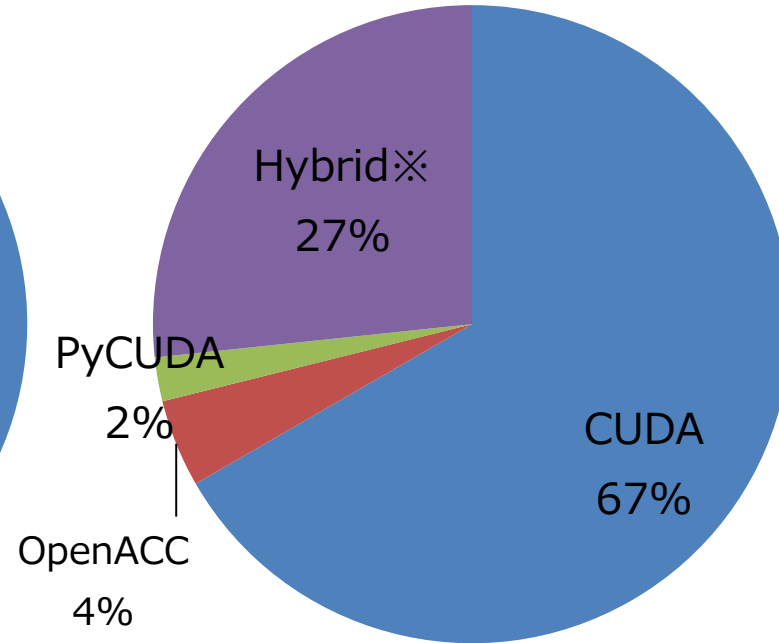
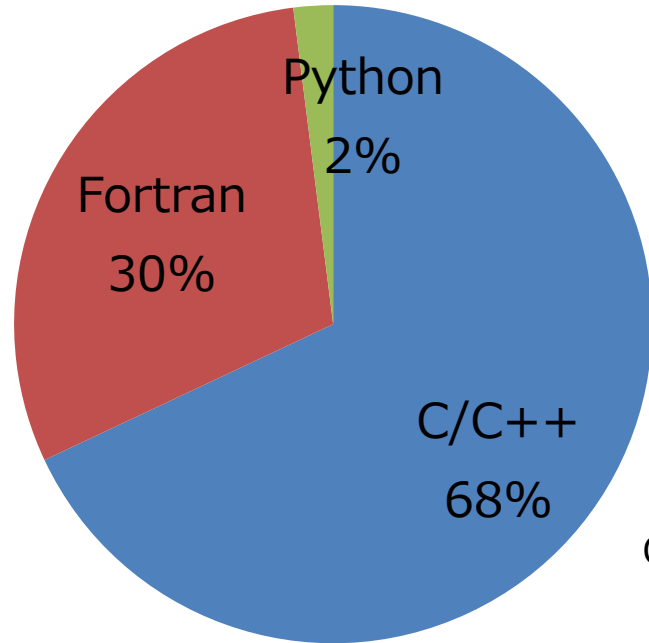
■ メモリアクセス

効率の良いメモリ配置

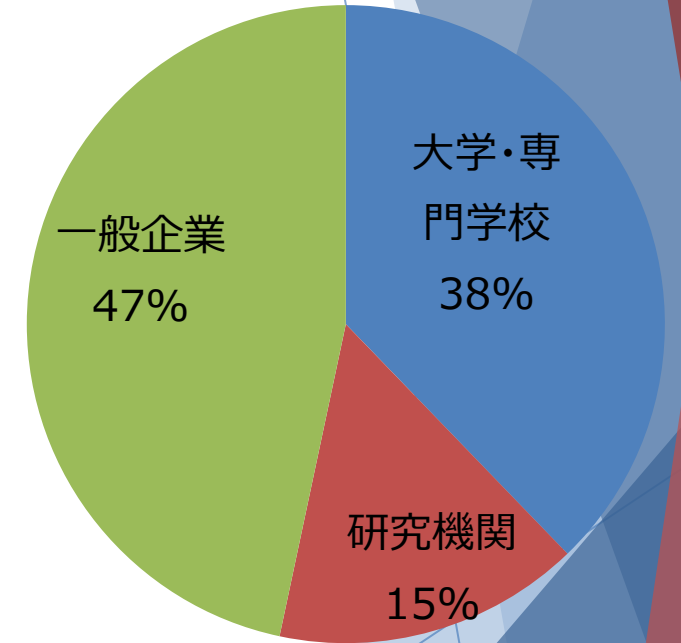
- ▶ CUDA の実行モデルの場合、コアレスアクセスを意識

事例紹介

導入実績 (2019年8月現在 : 50件)



※MPI+(CUDA/OpenACC)



事例紹介

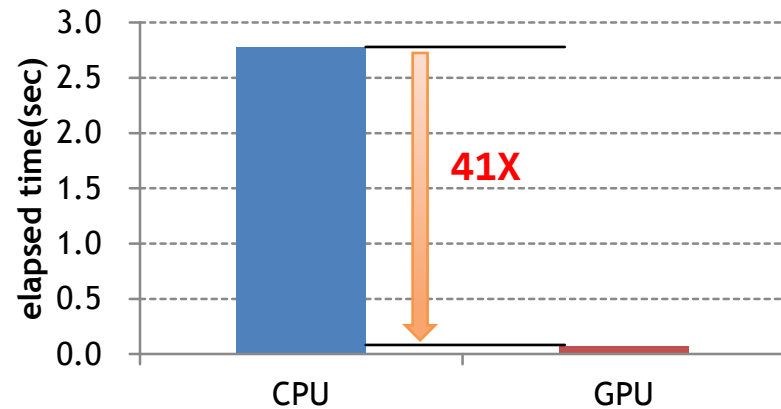
導入事例（2019年8月現在、一部抜粋）

No.	お客様	概要	言語	OS
1	A 大学	洪水氾濫モデルの高速化	CUDA C	Linux
2	D 研究所	大規模解析データの高速可視化	MPI+OpenMP	Linux
3	B 大学	大気物理過程計算の高速化	OpenACC	Linux
4	株式会社 B	顔画像解析計算の高速化	CUDA C	Windows
5	C 大学	大気乱流計算の高速化	OpenACC	Linux
6	株式会社 D	電線束形状計算の高速化	CUDA C+OpenMP	Windows
7	株式会社 F	超音波信号解析計算の高速化	CUDA C	Windows
8	E 研究所	分子動力学計算の高速化	CUDA Fortran	Linux
9	E 研究所	ScaLAPACK の GPU ライブラリ作成	CUDA C+MPI+OpenMP	Linux
10	株式会社 G	外れ値検出法(LOF法)の高速化	pyCUDA	Linux
11	E 研究所	モンテカルロ法の高速化	CUDA C+MPI+OpenMP	Linux
12	株式会社 C	CAE（溶接）解析計算のマルチ GPU 対応	CUDA C	Windows
13	株式会社 A	CAE（鋳造）解析計算の高速化	CUDA C	Windows

事例紹介

【事例 No.7】超音波信号解析計算

超音波信号の空間方向に伝播する速度を推測する
解析計算の高速化



【CPU】 Intel Core i7-3930K (1core)

【GPU】 NVIDIA Tesla K20c

【OS】 Windows7 Professional

【Language】 CUDA C

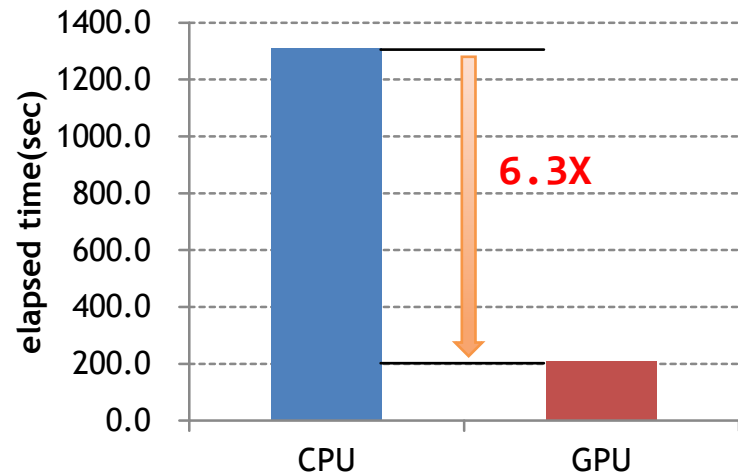
【高速化ポイント】

- 並列数
- データ転送

事例紹介

【事例 No.13】 CAE（鋳造）解析計算

鋳型内部への溶けた金属の湯流れや凝固過程をシミュレーションするパッケージソフトの高速化



【CPU】 Intel Xeon CPU E5-1650v4 (12core)

【GPU】 NVIDIA Quadro GP100

【OS】 Windows10 Proffesional SP1 64bit

【Language】 CUDA C

【高速化ポイント】

- 並列数
- データ転送
- メモリアクセス

並列高速化の研究

自動 OpenACC 化に向けた研究

- Fortran, C/C++
- ディレクティブ自動生成・最適化

GPU 対応ライブラリの検証

- cuBLAS, cuSPARSE 等
- MAGMA

GPU 性能評価

- アトミック演算
- データ転送

コンサルティングサービス

■ コンサルティングサービスとは？

- お客様のプログラムに対して、並列高速化をサポート
- CUDA (Fortran, C/C++), OpenACC はもちろん、MPI, OpenMP も対象
- 基本的には**お客様環境下**でコンサルティングを実施

コンサルティングサービス

このようなお客様におススメ！

- 自社開発したプログラムのため、**外部に公開出来ない※**が並列高速化を行いたい

※基本的に高速化専門の技術者が訪問し、プログラムを拝見させて頂くスタイルです

メリット

- プログラムを公開しなくてよいので、安全性が高い
- 直接自社開発したプログラムに対してアドバイスがもらえるので、より効果的
- 並列高速化についての疑問や問題点を直接聞くことが出来る

コンサルティングサービス

流れ

【工程 1】ヒアリング

- 現状の把握
- コンサルティングの進め方の確認

【工程 2】スケジューリング

- スケジュールの作成
- お見積り

【工程 3】コンサルティング開始

- お客様環境下でスタート
- 疑問点や問題点をクリアに

【スケジュール例】

第 1 回 お客様開発プログラムのレクチャーと確認

第 2 回 GPU アーキテクチャの説明

第 3 回 プロファイリング等による初期解析

第 4 回 メモリの取扱いについて

第 5 回 マルチ GPU の概要と導入是非について

第 6 回 さらに高速化にむけて

第 7 回 まとめ

ご清聴ありがとうございました