

# 解析プログラムの高度化・高速化

## プログラムの高度化

お客様の保有する解析プログラムの管理状態に合わせ、既存コードの構造化・API仕様作成等を実施し、プログラムの保守性・拡張性・可読性・信頼性の向上を実現します。

### 単体テストの実施

結果比較

自動テストツールによる単体テスト実行

### コーディング規約を利用したスタイルの統一

コーディング規約

### 構造化プログラミングの適用

### API仕様自動作成

自動生成

コメント文のルール化

## プログラムの高速化

パフォーマンス分析により既存コードのホットスポットを特定し、逐次プログラムの高速化および共有メモリ型・分散メモリ型並列化による高速化を実現します。

### パフォーマンス分析に基づく高速化

高速化前の負荷		高速化後の負荷	
関数	負荷	関数	負荷
func1	High	func1	Low
func2	Low	func2	Low
...			
func50	Low	func50	Low

A社システムのプログラム処理速度が**7.4倍\***向上

  

### 高速化による計算時間推移

B社システムのプログラム処理速度が**10倍\***向上

\*弊社検証実績

### ■ 特徴

- 構造化による解析プログラムの保守性・拡張性・可読性・信頼性の向上
- パフォーマンス分析に基づくコード最適化による高速化
- 共有メモリ型または分散メモリ型並列化による高速化

### ■ OS

- Windows、Linux

### ■ 開発言語

- Fortran、C/C++

### ■ DB/開発ツール

- Visual Studio、Eclipse
- Intel Parallel Studio

### ■ 知識・技術トピック

- プロファイラによるプログラム性能解析
- コード最適化による高速化
- 並列化 (OpenMP, MPI, GPU)
- プロファイラによる並列化性能解析
- ドキュメント自動生成ツール (Doxygen)