

## SAR 画像再生処理への並列プログラミング支援環境の適用検討

5L-4

和泉 秀幸<sup>†</sup> 佐々木 和司<sup>‡</sup> 村岡 史子<sup>‡</sup> 濱窪 眞紀<sup>‡</sup> 水野 政治<sup>†</sup> 渡辺 泰行<sup>‡</sup>

<sup>†</sup> 三菱電機 (株) 情報技術総合研究所 <sup>‡</sup> 三菱電機 (株) 鎌倉製作所

### 1 はじめに

近年では、WS や PC といった汎用の環境で、マルチスレッドプログラムの開発が可能になってきている。しかし、一般に、正しくかつ効率の良いマルチスレッドプログラムを開発することは難しく、プログラミング支援環境に対する期待が大きい。

我々は、汎用的な UNIX でのマルチスレッドプログラムを対象に、実行モデルに基づく並列プログラム開発支援環境 M-PPE(Model based Parallel Programming support Environment) の構築法を提案している (図 1 参照)。M-PPE の効果を確認するため、まず、リアルタイムシステムである監視・制御システムの操作端末 S/W を対象に適用して効果を確認した [3]。今回は SAR(Synthetic Aperture Radar: 合成開口レーダ) 画像再生処理を対象に適用を行っている。本稿では、適用の第 1 ステップとして実施した並列化検討と、簡易スレッドライブラリである SPL(Simple Parallel Library) [4] を使って試験的に並列化した結果について述べる。

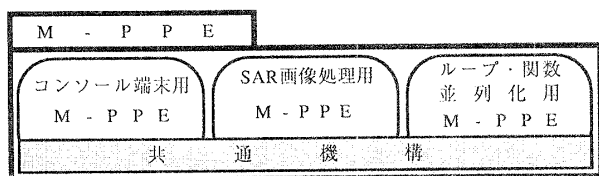


図 1 実行モデルに基づく支援環境

### 2 M-PPE

まず、M-PPE の構築方法を示す (図 2 参照)。

1. 開発対象のアプリケーションに適した並列処理機構を実行モデルとして規定。
2. 規定した並列処理機構を簡易ライブラリで実現。
3. 実現した簡易ライブラリを使用したアプリケーションを対象に、デバッグやチューニングの支援を行うツール群を開発。

開発者は M-PPE 提供のライブラリ I/F を利用することで、並列処理の制御に関する手続きをプログラム中に記述せずに、アプリケーションに適した並列処理機構を実現できる。また、M-PPE で規定した実行モデル

Applicability of Execution Model based Multithread Programming Support Environment for SAR processing  
Information Technology R&D Center, Mitsubishi Electric Corporation, 5-1-1 Ofuna, Kamakura, Kanagawa, 247-8501, Japan

を複数の開発者間での共通基盤にすることで、ロック獲得・開放の不整合などを減少でき、デバッグ効率を高めることができる。

M-PPE を用いて開発したアプリケーションを対象に性能評価を行うため、性能評価やデバッグを行うツールでは、対象の実行モデルでポイントとなる計測箇所と情報を明確にできる。このため、必要な情報を自動的に収集できる。また、自動収集した情報を使って効率の良い性能評価や、再現実行機能などの実現が期待できる。

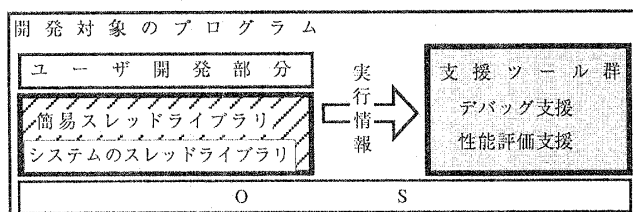


図 2 M-PPE によるプログラムの開発

次に、M-PPE を使ったマルチスレッドプログラムの開発手順を示す。

1. 開発対象プログラムとスレッド機能の調査  
開発対象プログラムと、実現対象システムのスレッド化機能を調査。
2. プロトタイプ作成と評価  
プロトタイプを作成し、並列処理機構の適用可能性を評価。
3. 実行モデル決定  
プロトタイプの評価結果から、並列処理機構を選択し、実行モデルとして規定。
4. プログラム開発と支援環境の構築  
まず簡易スレッドライブラリを開発し、その後、対象アプリケーションとデバッグ、性能評価・チューニング用の支援ツールを並行開発。

### 3 SAR 画像再生処理の並列化検討

SAR は、日中、夜間、雲霧等の天候を問わずに、高い分解能で地表を撮像できるセンサであり、人間が理解可能な画像を生成する再生処理が必要になる。この SAR 画像再生処理では、収集した“SAR シグナルデータ”(生データ)を信号処理して画像を得るが、演算量が多いため、並列処理等による高速化が必要になる。

この画像処理高速化の手段として、マルチスレッドプログラムによる並列処理を検討している。今回は、M-PPE の構築法を適用して検討を行った。

### 3.1 SAR 画像再生処理の解析

適用検討として、まず、プロトタイプで評価する並列処理機構の候補を選定することを目的に、SAR 画像再生処理の解析を行った。ここでは、並列可能部分を抽出するため、各部分処理毎に、各部分処理での並列化の可否と、部分処理間の依存関係を解析した。その結果を図3に示す。なお、SAR 画像の再生方法の詳細については、説明を省略する [5]。

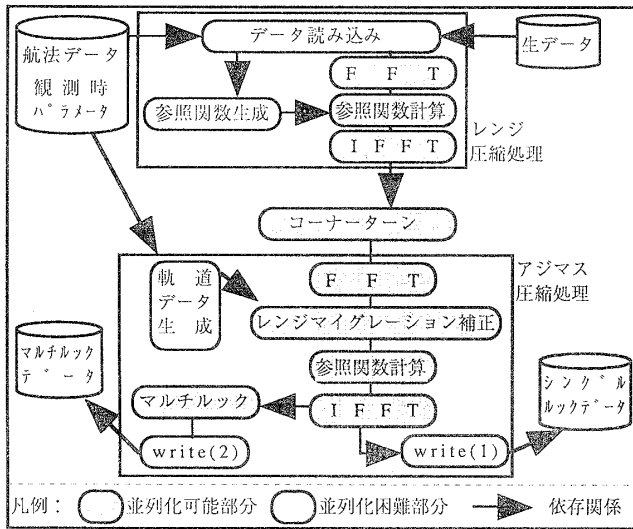


図3 SAR 画像再生処理の解析

SAR 画像再生処理では、各処理部分で画像内の処理結果に依存関係が少なく、並列処理可能な部分が多い。一方で各処理部分での結果を、次の処理部分で利用する依存関係があり、処理部分単位での並列化は難しい。

### 3.2 スレッドによる並列化方法の検討

並列化の可否と依存関係を解析した後、各処理部分での処理負荷を解析した。この結果、レンジ圧縮処理とアジマス圧縮処理で、それぞれ“FFT から IFFT までの処理”が90%以上と特に高いことがわかった。また、処理画像の書き込みなどの I/O 部分の処理負荷が高いと予測された。

このため、共有メモリの特性を活かして、処理前と処理結果のメモリ領域を確保し、各処理部分での画像処理を並列実行する並列処理機構が有効であると考えた(図4参照)。この処理機構では、画像処理を並列実行できるだけでなく、処理前のメモリ領域を利用して、I/O 処理と画像処理を並列に実行することも可能である。

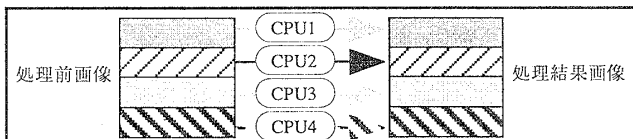


図4 並列化方法

### 4 並列化プロトタイプ作成と性能測定

考案した並列処理機構を評価するため、プロトタイプ の作成を行った。処理負荷が高い“FFT から IFFT までの処理”は、主に信号処理であり、ループの並列化を応用して簡単に実現できる。

このため、簡易スレッドライブラリ SPL[4] を使って、マルチスレッドプログラムを試作した。SPL は、他の信号処理システム向けに開発した簡易スレッドライブラリであり、ループや関数の並列化機能がある。

試作では、最も処理負荷の高いアジマス圧縮処理を選択して、実行速度を計測した。その結果を図5に示す。なお、計測環境は、Sun Enterprise 450, Ultra SPARC II(300MHz) × 4CPU である。

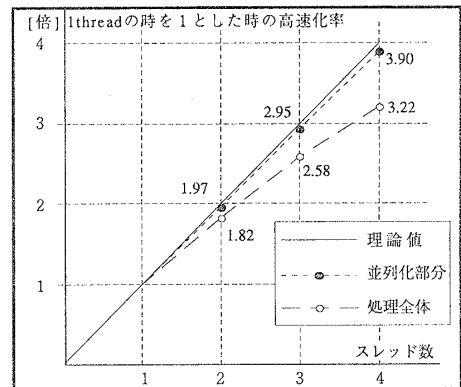


図5 アジマス圧縮処理の実行結果

### 5 まとめ

SAR 画像再生処理への M-PPE 適用の第1ステップとして並列化検討を実施し、並列化が可能な部分を抽出した。また、並列処理機構として、処理前と処理結果のメモリ領域を使って並列実行する方法を考えた。

この並列処理機構の効果を検証するため、SPL を使って SAR 画像再生処理を部分的に並列化し、実行時間を計測した。この結果、4CPU を使って、並列化部分で3.9倍、処理全体で3.2倍の高速化が確認できた。

### 参考文献

- [1] Ziya Aral and Ilya Gertner: *Non-intrusive and interactive profiling in Parasight*, ACM SIGPLAN NOTICES, 23(9):21-30,1988.9.
- [2] Ziya Aral and Ilya Gertner: *High-level debugging in Parasight*, ACM SIGPLAN Notices, 24(1):151-162,1989.1.
- [3] 和泉 秀幸、中島 毅: 実行モデルに基づく並列プログラミング支援環境の構築, 本会研究会 SE-120-13, pp.85-92, 1998
- [4] 福地 雄史、石塚 章子、和泉 秀幸: マルチプロセッサ対応 UNIX 上での並列プログラム開発支援環境の開発, 第 48 回 情処全国大会 2G-9, 1994.3.
- [5] 飯坂讓二監修 日本写真測量学会編: 合成開口レーダ画像ハンドブック, 朝倉書店,1998.5.