

# ユビキタス並列アプリケーション用プロトタイピング環境の開発

- 組み込みアプリケーションを気軽に作れる環境 -

## 1. 背景

組み込み機器の高性能化および普及の状況を鑑みると、次世代ユビキタス環境においては身近に存在する組み込み機器の分散・協調並列処理により様々なユビキタスアプリケーションを実現することが可能ではないかと考えられる。このような観点から、開発者は組み込み機器による分散・協調並列処理の可能性を模索するためのプロトタイピング環境およびアプリケーションの開発を試みてきた。ハードウェア的な環境の整備が充実してきた一方で、多くの組み込み機器では小数演算を伴うアプリケーションの高速処理が技術的あるいは開発期間の観点から困難であるという問題が明らかとなった。

## 2. 目的

組み込み型分散・協調並列処理システムにより、小数演算を伴うアプリケーションを高速処理するためのプログラムを短期間にかつ容易に開発することが可能となるような基盤ソフトウェアを開発し、ソフトウェア的にもプロトタイピング環境を充実させることが本テーマの目的である。

## 3. 開発の内容

本テーマにおける開発内容は下記の通りである。

- ・ 基盤ソフトウェア評価用プラットフォーム
- ・ 基盤ソフトウェア: 固定小数点演算関数ライブラリ
- ・ 基盤ソフトウェア: 画像照合関数ライブラリ

以下に詳細な開発内容を示す。

### 基盤ソフトウェア評価用プラットフォーム

開発した基盤ソフトウェアの精度および速度性能を評価するためのプラットフォームとして、マルチコア組み込み CPU ボードを分散・協調並列用クラスタ計算機を開発した。図 1 および表 1 にそれぞれ実機写真と詳細仕様を示す。



図 1 開発したクラスタ計算機『M32RUCC』

表 1 クラスタ計算機および計算ノードの仕様

クラスタ計算機の仕様	
ノード数	4 (8)
ネットワークインタフェース	100Mbps高速イーサネット(ネットワークHUBを搭載)
周辺機器インタフェース	USB 2.0(12Mbps Fullspeed) × 2port
OS	Debian / GNU Linux 2.6
並列処理インタフェース	LAM / MPI version 7.1.2
開発環境	GNU C, C++, FORTRAN version 3.4.3
外形寸法	288mm(W) × 224mm(D) × 128mm(H)
電源	5V/13A (65W)
計算ノードの諸元	
CPU	M32R CPU (デュアルコア, @300MHz)
メインメモリ	128MB SDRAM
HDD	4GB (MicroDrive)
ネットワークインタフェース	100Mbps高速イーサネット
周辺機器インタフェース	USB 2.0(12Mbps Fullspeed) × 2port
電源	5V/3A

### 固定小数点演算関数ライブラリ

本テーマでは、小数を扱うアプリケーションを高速処理するための手段として、C 言語の標準関数と同等に扱える固定小数点演算関数ライブラリを提供することを提案し、この開発を行った。開発期間が限られているため、開発対象を画像照合に基づくアプリケーションを実現するための関数に限定し、表 2 に示すような四則演算関数、三角関数(sin, cos, atan)、1次元および2次元フーリエ変換関数を開発した。また、C 言語標準の実数(float 型、double 型)と開発した固定小数点関数とのデータのやり取りを容易にするために、浮動小数点 固定小数点の型変換関数を開発した。関数ライブラリの特徴は下記の通りである。

- ・ C 言語の標準関数などと同様に扱うことができる。
- ・ 小数点位置を変数により管理することで、任意の固定小数点フォーマットに対応できる。
- ・ 演算速度評価結果から、C 言語の浮動小数点演算(関数)と比較して、数倍から 10 倍程度の高速化が達成されている。演算精度および速度の評価については、紙面の都合上、開発ホームページをご参照いただきたい。

表 2 開発した固定小数点演算関数一覧

関数名	機能	関数名	機能
Float2Fixed	型変換(float fixed)	FixedLookupSin	テーブル参照型sin関数
Double2Fixed	型変換(double fixed)	FixedLookupCos	テーブル参照型cos関数
Fixed2Float	型変換(fixed double)	FixedTschebSin	チェビシェフ近似型sin関数
Fixed2Double	型変換(float fixed)	FixedTschebCos	チェビシェフ近似型cos関数
FixedAdd	加算	CordicAtan	CORDIC型atan関数
FixedSub	減算	FixedFFTDIT	DIT型1次元フーリエ変換
FixedMul	乗算	FixedFFTSR	SR型1次元フーリエ変換
FixedMulHS	高速乗算	FixedFFT2DDIT	DIT型2次元フーリエ変換
FixedDiv	除算	FixedFFT2DSR	SR型2次元フーリエ変換

## 画像照合関数ライブラリ

上記固定小数点演算関数ライブラリを用いた画像照合関数ライブラリを開発した。表 3 に示すように、画像入出力(ビットマップ画像およびRAW画像)、画像の幾何学的変換(平行移動、回転移動、拡大縮小)、切り出し、照合の各機能を実現する関数を作成した。照合関数には位相限定相関関数に基づく手法を採用している。

また、速度評価用に同一アルゴリズムに基づく浮動小数点演算版の画像照合関数ライブラリを開発した。演算精度の評価結果より、固定小数点版画像照合関数ライブラリは、浮動小数点版と比較して高速化が達成されている。詳細な評価結果については、紙面の都合上、開発ホームページをご参照いただきたい。

表 3 開発対象とした画像処理・画像照合関数一覧

関数名(固定小数点版)	関数名(浮動小数点版)	機能
	RawRead	Rawフォーマット画像読み込み
	RawWrite	Rawフォーマット画像書き出し
	BmpRead	ビットマップ画像読み込み
	BmpWrite	ビットマップ画像書き出し
ImTranslateBW	FlmTranslateBW	白黒濃淡画像の平行移動
ImTranslateBW	FlmTranslateBW	カラー画像の平行移動
ImRotateBW	FlmRotateBW	白黒濃淡画像の回転移動
ImRotateBW	FlmRotateBW	カラー画像の回転移動
ImResizeBW	FlmResizeBW	白黒濃淡画像の拡大縮小
ImResizeBW	FlmResizeBW	カラー画像の拡大縮小
ImCropBW	FlmCropBW	白黒濃淡画像のクロッピング
ImCropBW	FlmCropBW	カラー画像のクロッピング
AutoCropBW	FAutoCropBW	白黒濃淡画像の簡易自動クロッピング
AutoCropBW	FAutoCropBW	カラー画像の簡易自動クロッピング
PhaseGeneratorBW	FPhaseGeneratorBW	白黒濃淡画像の位相計算
PhaseOnlyMatchingBW	FPhaseOnlyMatchingBW	白黒濃淡画像の画像照合

開発した固定小数点演算ライブラリと画像照合関数ライブラリにより画像照合アプリケーションを作成し、照合時間の評価を行った。開発したアプリケーションは、入力画像とデータベース画像とを繰り返し照合し、最大スコアを出力するものである。同様のアプリケーションをC言語の浮動小数点演算を用いて作成し、実行速度の比較を行った結果、10倍以上の高速化が達成されることを確認している(表 4)。

表 4 サンプルプログラムによるアプリケーション処理性能の評価

浮動小数点演算				
CPU	逐次処理	並列処理 (4ノード)	並列処理 (8ノード)	並列処理(4ノード) + マルチスレッド
M32R	204.17 [sec]	57.71 [sec]	32.79 [sec]	33.16 [sec]
ARM9	320.35 [sec]	89.31 [sec]	-	-
固定小数点演算				
CPU	逐次処理	並列処理 (4ノード)	並列処理 (8ノード)	並列処理(4ノード) + マルチスレッド
M32R	9.22 [sec]	2.65 [sec]	1.61 [sec]	1.64 [sec]
ARM9	10.43 [sec]	2.93 [sec]	-	-

#### 4. 従来の技術(または機能)との相違

- ・ 本テーマ開始当時の調査の範囲では、オープンソースとして公開されている固定小数点演算ライブラリはほとんど存在せず、新たな試みといえる。
- ・ 存在する固定小数点演算ライブラリは、CPU メーカーが有償で提供するライブラリや、固定小数点フォーマットについて自由度の少ないものであるのに対し、開発した関数ライブラリはフリーで公開するものであり、また、固定小数点フォーマットはユーザがある程度自由に設定することが可能である。
- ・ メーカーが有償で提供するライブラリが機種依存してしまうのに対し、開発したライブラリ関数はソースコードで提供するため、Linux OS と C コンパイラが動作する環境であれば、機種依存することなく活用することができる。
- ・ 実機上で精度や速度に関する演算性能を評価しており、評価結果はホームページで公開されている。

#### 5. 期待される効果

- ・ 開発したライブラリは、一般的なC言語の関数と同様に利用でき、また、小数点位置を変数で管理できるため、組み込み機器において固定小数点演算によるアプリケーションプログラムの作成が容易となる。
- ・ 上記評価結果の通り、一般的な浮動小数点演算を用いた場合と比較して、関数レベルでは浮動小数点演算の数倍から10倍程度、アプリケーションレベルでは数10倍程度の高速化が達成されている。すなわち、比較的計算能力の低い組み込み機器でも、小数を扱うアプリケーションの実時間処理が期待できる。

#### 6. 普及(または活用)の見通し

具体的な数値を挙げることは難しいが、ライブラリをWeb公開し、組み込み機器による高速処理に興味を持っているできる限り多くの人々に活用してもらい、また、本テーマの上位の目的である組み込みクラスタ計算機システムによる高度なユビキタスアプリケーションの実現という概念が広く普及されることを期待する。

#### 7. 開発者名(所属)

佐々木慶文 (石巻専修大学理工学部情報電子工学科)

(参考)開発者URL

<http://hyperion.ie.isenshu-u.ac.jp/ecc/>