ベクトル（VPP5000）からスカラSMP（PRIMEPOWER HPC2500）へ

青　木　正　樹

Ⅰ．はじめに

高性能計算機を用いた計算科学は，バイオインフォマティクス，気象予測，構造解析など，学術・産業の幅広い分野で大きな成果を上げています。より高精度・大規模な計算を実現するために，より高性能な計算機が求められています。

PRIMEPOWER HPC2500（以降，HPC2500と略す）は，これらのニーズに応えるために富士通が開発したHPC（High Performance Computing）サーバです。

HPC2500は，VPPシリーズのベクトルパラレル処理技術（以降，VPP技術と略す）とUNIXサーバPRIMEPOWERのSMP（Symmetric　Multi Processor）技術を融合するHPCサーバであり，最大128個のスカラCPUから成るSMPノードを，高速光インタコネクト装置を用いて最大128台結合することで，世界最大級の並列マルチノードシステムを構成します。

（VPP技術は，プログラミングの容易さ・並列処理効率の高さ・適用アプリケーションの広さで定評のある並列処理技術であり，PRIMEPOWERは，メインフレームコンピュータ技術を駆使した，世界最高レベルの高性能・高信頼性・拡張性を実現するUNIXサーバです）

名古屋大学情報連携基盤センターでは，2005年春，スーパーコンピュータシステムをVPPシリーズからHPC2500にリプレースします。図１に移行の概略図を示します。本稿では，VPPシステムとHPC2500システムの構成と並列処理方式について紹介し，さらにVPPシリーズからHPC2500へ移行する際のチューニングのポイントについても解説します。

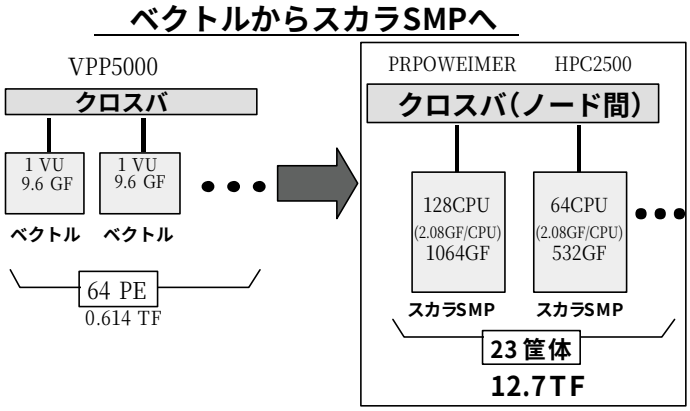


図１　ベクトルからスカラSMPへ

Ⅱ．HPC2500のシステム構成

HPC2500システムの構成を図２に示します。

８個のスカラCPU，メモリ，及び基本IOアダプタを搭載したシステムボードと，ノード間データ転送を行うためのDTU（Data Transfer Unit）ボードを，高速なクロスバスイッチで接続してノードを構成します。ノードには最大16枚のシステムボードと最大２枚のDTUボードを接続することができ，最大構成時には128CPU，メモリ容量が512Gバイトの SMPとなります。

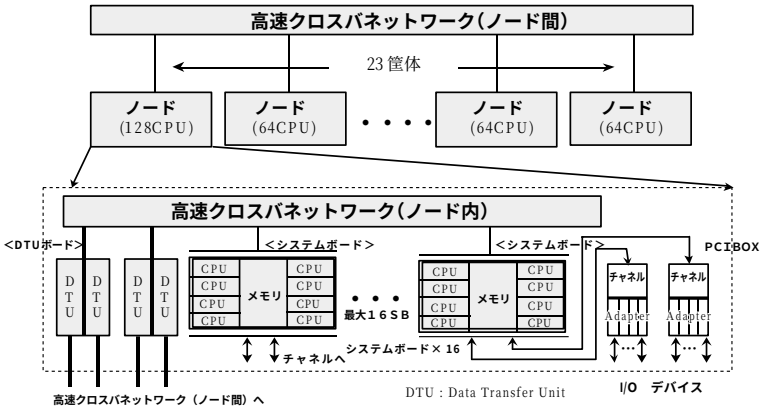


図２　HPC2500のシステム構成図

Ⅲ．高速化技術の比較

HPC2500はVPPシステムにおけるベクトルプロセッサエレメント（PE）に対応してスカラプロセッサのSMPノードを配置し，システム構成上の類似性を実現します。クロスバネットワークのアクセス方式もVPPシステムと同様の方式を採用しています。したがってプログラム開発上，図３の高速化技術の対比図に示すように，ユーザからのシステムの見え方が共通であり，ベクトルジョブ，ベクトル並列ジョブ資産の継承が図れます。

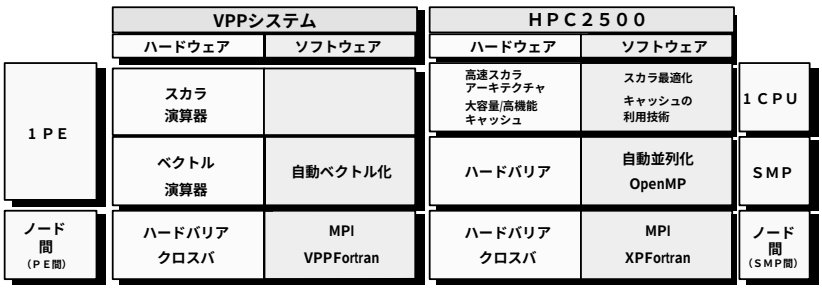


図３　高速化技術の対比（VPPとHPC2500）

VPPシステムでは，大容量のデータ演算をベクトル演算器で実行することで，高速性能を実現しました。

キーテクノロジは，『ベクトル演算器』とコンパイラの『自動ベクトル化』技術です。

HPC2500では，スーパスカラ技術（out of order実行，演算パイプライン化，多重命令実行など），大容量/高機能キャッシュと，これらのハードウェア能力を引き出すコンパイラ最適化（命令スケジューリング，データプリフェッチによる大容量データの高速アクセス）により，高速のスカラ性能を実現します。SMP高速化に対しては，並列化オーバーヘッドを小さくする高速ハードウェアバリア機能の活用と，VPPでの自動ベクトル化技術を応用/発展させた自動並列化技術/OpenMPの適用により高速化を実現しています。

ノード間高速化に対しては，VPPで培った高速ハードウェアバリア/DTUと，高速ライブラリ（MPI/XPFortran）の技術を継承・発展させています。

さらに，HPC2500では，VPPシステムと比較して，自動並列化・高速化コンパイラの適用範囲が大きく拡大しました。VPP5000では１PEのメインメモリサイズは16Gバイトで，データサイズがこれを超えなければ自動ベクトル化コンパイラを用いて容易に高速処理を行えます。しかし，より大きなデータサイズの処理を実行するためには，PE間並列処理のためのプログラム書き換え・並列化指示行の追加が必要となります。

HPC2500において自動並列化・高速化の対象となるノードのピーク性能はVPP5000の１PEの約55倍（ノードあたり64CPU構成機の場合），メインメモリ容量は32倍です。コンパイラによる自動並列化・高速化の適用範囲は，性能，データサイズの両面で飛躍的に拡大しています。プログラムの並列化書き換え・並列化指示行の追加作業がネックとなり大規模高速処理を断念していたユーザでも，先進のノード内自動並列化コンパイラを用いて容易に大規模高速処理を行うことができます。

Ⅳ．プログラムの移行

VPPからHPC2500に移行する際，図３の対比のようにベクトル処理に対してスレッド並列処理を対応させるのが基本的な考え方です。VPPで自動ベクトル化を適用していたプログラムは，HPC2500では自動並列化（必要に応じて最適化制御行も使用できます）あるいはOpenMPディレクティブを利用してスレッド並列処理を適用してください。

VPPでベクトル処理とプロセス並列処理（PVM,MPI及びVPP Fortran）を併用できたように，HPC2500でもスレッド並列処理とプロセス並列処理を併用することができます。

図４にVPPからのプログラムの移行例を示します。

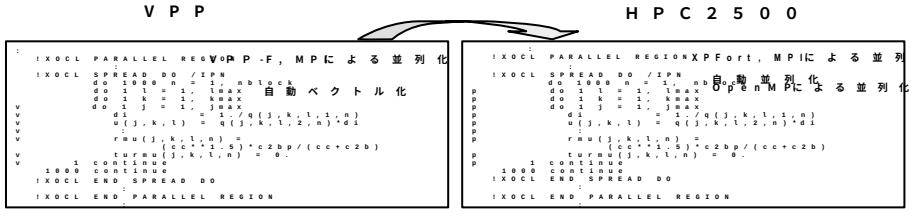


図４　VPPからHPC2500へのプログラム移行例

ただし，上記は基本的な考え方であって，この対応にとらわれずスレッド並列のみあるいはプロセス並列のみを使用することも可能です。

１．ソースプログラム資産

VPPシステム上で動作している，Fortran,C/C++言語で記述されたソースプログラムについては，再翻訳することで再利用が可能になります。VPP Fortranプログラムについては，VPP Fortran言語仕様を包含したXPFortran言語仕様をサポートしており，ソースプログラムについては再翻訳することで再利用が可能になります。HPFプログラムについては，HPFコンパイラをサポートしていないため，再翻訳するだけでは移行することができません。図５に言語処理系の対比図を示します。

	VPP	HPC2500
逐次言語	Fortran，C，C++ 自動ベクトル化	Fortran，C，C++ 自動並列化
データ並列言語	VPP Fortran，HPF	OpenMP，XPFortran
ツール	ANALYZER VPP Workbench MPTools	Parallelnavi Workbench
メッセージパシング ライブラリ	MPI，PVM	MPI
数学ライブラリ	SSL11，C-SSL11 BLAS，LAPACK ScaLAPACK	SSL11，C-SSL11 BLAS，LAPACK ScaLAPACK

図５　言語処理系の対比

２．浮動小数点演算の計算結果

VPPシステムとHPC2500双方ともIEEE754に準拠していますが，以下の要因により計算結果に違いが生じることがあります。

１）コンパイラの最適化機能の違い

２）数学関数の内部アルゴリズムの変更による精度差

３）非正規化数があらわれた場合の処理の違い（コンパイラの-Knsオプション指定）

４）総和演算命令（VPPのみ存在）使用による精度差

５）mult&add命令による精度差

３．その他

１）CLOCKVサービスサブルーチン

CLOCKVサービスサブルーチンの第一引数（VU時間をあらわす）には必ず０が返却されます。

2）組込み演算と浮動小数点演算のエラー検出

HPC2500では-NRnotrapオプションがデフォルトですが，VPPでは-NRtrapオプション指定相当の動作でした。HPC2500でVPPと同等の動作は-NRtrapオプションを指定してください。

3）組込み手続きの補正値

組込み手続でエラーが発生した場合の修正解釈値が変更されています。この修正値での動作にソースプログラムを修正してください。

4）-Mオプションの指定

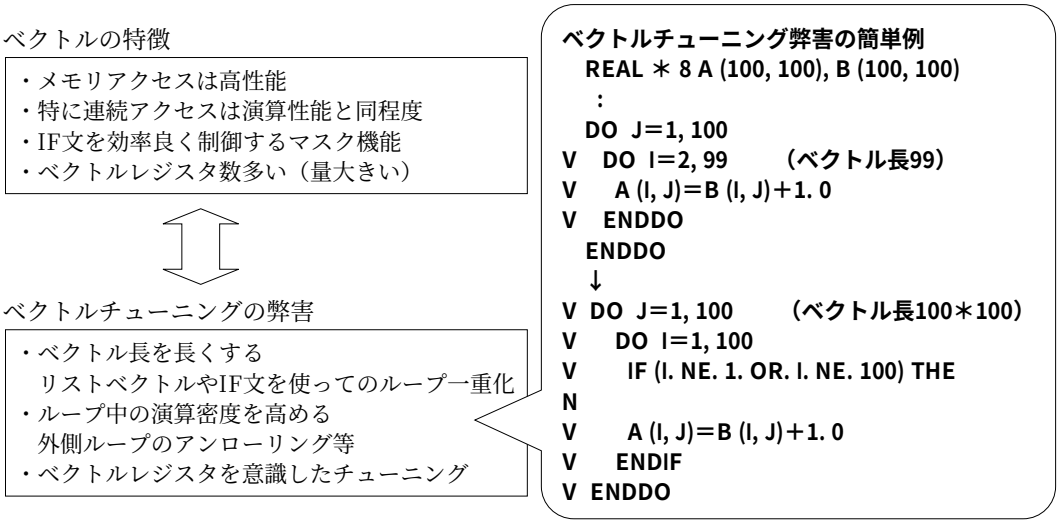
翻訳時オプション-Mは，-Amオプションと組合せて指定しなければいけません。

Ⅴ．ベクトルからHPC2500への移行する場合のチューニングポイント

ここでは，VPPでチューニングされたプログラムをHPC2500へ移行する際のチューニングのポイントを簡単にまとめます。

１．ベクトルの特徴とベクトルチューニングの弊害

ベクトル演算器の特徴を活かしたベクトルチューニングは，HPC2500にとって弊害（実行性能の劣化）を及ぼす可能性があります。



２．HPC2500活用の基本

基本的なスカラ並列計算機活用のポイントを示します。

スカラ並列計算機の特徴
・多CPU
・マシンクロックが速い
・メモリアクセスに階層構造がある　キャッシュ（１次←２次）←メモリ
相対速度　1　10　100

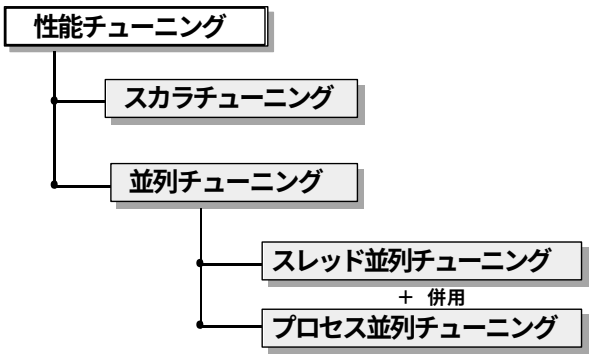
スカラ並列チューニングのポイント
・並列化（多数CPUの利用）
・メモリアクセスの局所化（キャッシュの有効活用）
・効率よい演算

Ⅵ．HPC2500チューニングのポイント

HPC2500での性能チューニングのポイントをまとめます。

まず，性能チューニングには，スカラチューニングと並列チューニングの二種類があり，さらに，並列チューニングには，スレッド並列チューニングとプロセス並列チューニングの２種類があります。

性能チューニングの体系



１．スカラチューニングのポイント

スカラ性能を向上させる上で考慮すべきポイントを以下に挙げます。

（１）データの局所性を高める

一般的にスカラ計算機では，プロセッサと主記憶の間に限られた容量の高速なメモリ（キャッシュメモリ）を用意して，それを階層的に組み合わせることにより，メモリアクセスを高速化させます。

そのため，アクセスするデータの局所性を高めることでキャッシュメモリに対してのヒット率を向上することが可能となり，メモリアクセスの高速化を図ることができます。

（２）演算器の実行効率を高める

SPARC64Vプロセッサの場合，１マシンサイクルあたり４命令の同時発行が可能であり，この命令の並列動作特性を最大限に生かし，演算密度を高めることが高速化につながります。

２．並列チューニングのポイント

並列性能を向上させる上で考慮すべきポイントを以下に挙げます。

（１）並列化率を上げる

並列化率をnとした場合に，アムダールの法則によると∞CPU数で並列化したとしても100÷（100-n）となるため，並列化率50%では，並列化効果はMAXで２倍にしかなりません。

実行コストの高いループはすべて並列化を行う必要があります。

アムダールの法則（Amdahl's law）

プログラム全体の実行時間のｐ（割合：０≦ｐ≦１）にあたる部分をｎ倍向上させた場合の，プログラム全体の性能向上比は以下のとおり。

$$\text{性能向上比} = \frac{1}{(1-p)+\frac{p}{n}}$$

【例】

・プログラム全体時間の90%にあたる処理を10倍性能向上　→　性能向上比＝約5.26倍

・プログラム全体時間の30%にあたる処理を10倍性能向上　→　性能向上比＝約1.37倍

図６　アムダールの法則

（２）並列化粒度を大きくする

並列化のオーバヘッドには，並列化を行うための処理のオーバヘッド，及び同期処理のオーバヘッドがあります。並列化粒度が小さい場合，このオーバヘッドが相対的に大きく見え，並列化効果を阻害します。

十分大きな並列化粒度とすることにより，このオーバヘッドを相対的に小さくすることが可能となります。

（３）CPU負荷バランスを均等化する

並列化の際には個々のCPUで請け負う仕事量を均等化すべきです。均等化できなかった場合，ある特定のCPUの負荷が高くなり，CPU台数効果が期待できなくなります。

（４）スレッド間のキャッシュ競合を回避する

SMPでのスレッド並列化の場合，メモリアクセス競合（キャッシュの奪い合い）が発生することが多々あります。HPC2500のキャッシュラインは64バイトであり，この64バイトに対して別々

のCPUが並列に書き込みを行った場合，大きな性能低下となります。キャッシュ競合を回避するためには，配列宣言の変更/PAD化や並列化粒度を大きくすることにより競合するメモリを少なくするなどの方法があります。

（５）プロセス間の通信コストを削減する

プロセス並列の場合，プロセス間の通信時間は，並列化のオーバヘッドとして見えてきます。少量データで通信回数が多い場合にはこのオーバヘッドが相対的に大きく見え，並列化効果を阻害します。通信の際のデータを大きくまとめ通信回数を削減すること等により，このオーバヘッドを相対的に小さくすることが可能となります。

Ⅶ．むすび

本稿では，ベクトルとHPC2500の対比を中心にシステム構成，並列処理方式，移行上の注意点及びチューニングのポイントを紹介しました。本稿が，大容量メモリかつ高ピーク性能を持つHPC2500の有効活用のヒントになれば幸いです。なお，ご質問，ご要望等あればセンターを通じご遠慮なく富士通株式会社の方に申し出てください。

以上

（あおき　まさき：富士通株式会社）
（m-aoki@jp.fujitsu.com）