SR2201 から SR8000/MPP への移行上の注意事項

（株）日立製作所

SR8000/MPP は、SR2201 の後継機であり、SR2201 上でご利用いただいているプログラムの多くは変更することなく、原始プログラムを SR8000/MPP 上でリコンパイルするだけで移行できます。しかし、SR8000/MPP では、浮動小数点データーの非正規化数に対する扱いの差及び、64 ビットアドレッシングモード、SR8000 互換の要素並列化、MPI-2 等の新規追加機能、仕様・性能改善があり、若干のプログラム修正が必要な場合があります。

SR8000 からの移行については、SR8000/MPP の OS、コンパイラー、ライブラリーは SR8000 と共通化しており、上位互換を有していますので、移行上の注意事項はありません。

以下では、SR8000/MPP の SR2201 に対する仕様差及び新規追加機能を説明するとともに、SR8000/MPP への移行上の注意事項につきましてご説明いたします。

1. FORTRAN 処理系

SR2201 で作成したプログラムを SR8000/MPP で実行するには、SR8000/MPP 上の FORTRAN コンパイラーで再コンパイルする必要があります。ただし、SR2201 で作成した原始プログラムをそのまま SR8000/MPP FORTRAN でコンパイルすると、ターゲットとするホストコンピューターのアーキテクチャーの違いが存在するため、実行時にエラーメッセージが出力されることがあります。

ここでは、SR2201 から原始プログラムを移行するときの注意事項について説明致します。

1.1 移行上の注意事項

SR2201 処理系固有の機能であり、SR8000/MPP では意識しなくてよいサービスサブルーチン及びコンパイルオプションについて以下に説明致します。

(1) サービスサブルーチン

①HF_DENORM ： 非正規化数の処理方法指定ルーチン

SR8000/MPP は、常に非正規化数を処理するため、本サービスサブルーチンは使用できません。

②HF_CFUNC ： FORTRAN から C 関数を呼び出すサービスサブルーチン

SR8000/MPP FORTRAN は、デフォルトで FORTRAN から C 関数を直接呼び出すことができるので、本サービスサブルーチンは使用できません。

③f_fcall ： C から FORTRAN 副プログラムを呼び出すサービスサブルーチン

SR8000/MPP C は、デフォルトで C から FORTRAN 副プログラムを直接呼び出すことができるので、本サービスサブルーチンは使用できません。

SR8000/MPP FORTRAN または C で、これらのサービスサブルーチンを原始プログラム中で引用している場合は、リンク時にエラーとなります。ただし、リンク時に未サポートサービスサブルーチンインフォメーション出力機能を使用すると、SR8000/MPP でこのサービスサブルーチンを実行した場合に、最初の 1 回目の呼び出し時に”本処理系では当該サービスサブルーチンが使用できない”旨のメッセージを出力し、2 回目以降の呼び出しではメッセージ

出力は行わずに実行を続けます。未サポートサービスサブルーチンインフォメーション出力機能につきましては、SR8000/MPP FORTRAN 対応マニュアル「HI-UX/MPP 最適化 FORTRAN77 使用の手引き」及び「HI-UX/MPP 最適化 FORTRAN90 使用の手引き」を参照してください。

(2) コンパイルオプション

①OPT(FOLD(1|2)) ： プログラム中のデーター参照時のベースアドレスに対する定数加減算の最適化指定。

SR8000/MPP では、本最適化実施時の副作用が存在しないため、最適化オプション指定時には常に本最適化が行われます。そのため、本コンパイルオプションを使用しても無視されます。

②-s,DENORM|NODENORM ： 非正規化数の処理方法指定

SR8000/MPP は、常に非正規化数を処理するため、本コンパイルオプションを使用しても無視されます。

(3) 実行時オプション

①RUNST(FPRECNTL(0|1))

SR8000/MPP FORTRAN では、ハードウェア例外処理の実行性能に対する影響を最小とするため、オーバーフロー及び除算例外を受け付けない状態をデフォルトとしています。SR2201 FORTRAN とは次に示す違いがありますので、注意が必要です。

・SR2201 FORTRAN ：オーバフロー及び除算例外を受け付ける。
・SR8000/MPP FORTRAN ：オーバフロー及び除算例外を受け付けない。

(4) *VOPTION 指示文

擬似ベクトル化を推進するための*VOPTION 指示文は SR8000/MPP FORTRAN でもそのまま有効となります。

(5) 手続き名称、COMMON 名称、MODULE 名称

次に示す名称は FORTRAN 文法上、実行可能プログラム内で大域的な名称ですので、プログラム単位間で重複して使用できません。SR2201 FORTRAN ではそのまま動作していましたが、SR8000/MPP FORTRAN ではリンク時にエラーとなります。

・COMMON ブロック名称
・サブルーチン名称
・MODULE 名称
・外部関数名称
・主プログラム名称
・初期値設定副プログラム名称

1.2　データー互換

SR8000/MPP は、数値データーを SR8000 形式で表現しています。しかし、他の FORTRAN 処理系では、数値データーを M 形式（M シリーズ処理装置で扱う形式であり、VOS3、HI-OSF/1-MJ システムで使用）又は SR2201（3050RX）形式で表現しています。

ここでは、SR8000 形式、M 形式、及び SR2201（3050RX）形式とのデーター互換について説明致します。なお、SR8000 形式と SR2201 形式の単精度浮動小数点形式と倍精度浮動小数点形式は同一であり、共に IEEE（Institute of Electrical and Electronic Engineer:アメリカ電気電子学会）形式に準拠しています。

(1) データー形式

SR8000 形式、M 形式、及び SR2201（3050RX）形式では、浮動小数点の表現範囲と精度が異なります。したがって、浮動小数点を使用しているプログラムの実行結果については、各システム間で誤差が発生する場合があります。

①浮動小数点の表現

SR8000 形式、M 形式、及び SR2201（3050RX）形式では、浮動小数点で表現できる数値の範囲に違いがあります。それぞれの形式で表現できる浮動小数点の数値の範囲を、表 1-1 に示します。

表 1-1 に示すように、指数の表現範囲は次のようになります。

・単精度浮動小数点

M 形式＞SR8000 形式＝SR2201（3050RX）形式

・倍精度浮動小数点

SR8000 形式＝SR2201（3050RX）形式＞M 形式

・拡張精度浮動小数点

SR2201（3050RX）形式＞SR8000 形式＞M 形式

このため、M 形式の単精度浮動小数点を使用したプログラム、又は SR2201（3050RX）形式の拡張精度浮動小数点を使用したプログラムを、SR8000/MPP で実行すると、オーバーフロー又はアンダーフローが発生する場合があります。

表 1-1　浮動小数点の表現範囲

精度	形式		
	SR8000 形式	M 形式	SR2201（3050RX）形式
単精度浮動小数点数	$\pm 1.175495\times10^{-38}$～$\pm 3.402823\times10^{38}$	$\pm 5.397606\times10^{-79}$～$\pm 7.237005\times10^{75}$	SR8000 形式と同じ
倍精度浮動小数点数	$\pm 2.225074\times10^{-308}$～$\pm 1.797693\times10^{308}$		SR8000 形式と同じ
拡張精度浮動小数点数	$\pm 2.225074\times10^{-308}$～$\pm 1.797693\times10^{308}$		$\pm 3.37\times10^{-4932}$～$\pm 1.189731\times10^{4932}$

②浮動小数点の精度

SR8000 形式、M 形式、及び SR2201（3050RX）形式では、浮動小数点の仮数部の有効ビット数に違いがあります。浮動小数点の仮数部の有効ビット数の違いを、表 1-2 に示します。

表 1-2　仮数部の有効ビット数

精度	形式		
	SR8000 形式	M 形式	SR2201（3050RX）形式
単精度浮動小数点数	24	21～24	SR8000 形式と同じ
倍精度浮動小数点数	53	53～56	SR8000 形式と同じ
拡張精度浮動小数点数	106	109～112	113

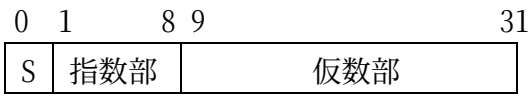
表 1-2 で示すように数値の精度がハードウェアによって異なるため、精度に敏感な計算を繰り返す場合は、システム間で実行結果に誤差が発生する場合があります。

③浮動小数点データーの形式

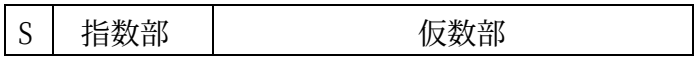
以下に、それぞれのデーターの形式を示します。

●SR8000 形式

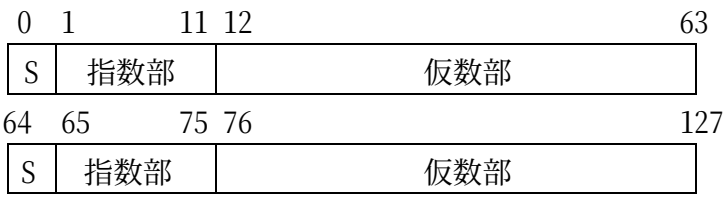
・実数型 4 バイト（単精度浮動小数点）の形式



・実数型 8 バイト（倍精度浮動小数点）の形式



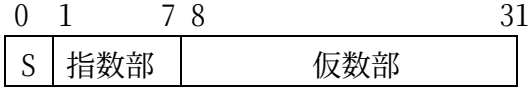
・実数型 16 バイト（拡張精度浮動小数点）の形式



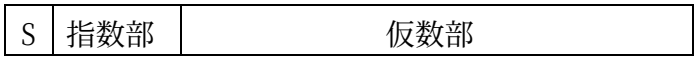
拡張精度の形式では、二つの連続した倍精度浮動小数点形式によって値を表現します。すなわち、ビット位置 0～63 では、浮動小数点数値の上位部分を表し、ビット位置 64～127 では浮動小数点数値の下位部分を表します。

●M 形式

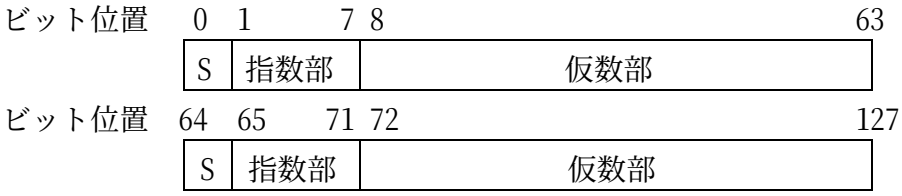
・実数型 4 バイト（単精度浮動小数点）の形式



・実数型 8 バイト（倍精度浮動小数点）の形式



・実数型 16 バイト（拡張精度浮動小数点）の形式



●SR2201（3050RX）形式

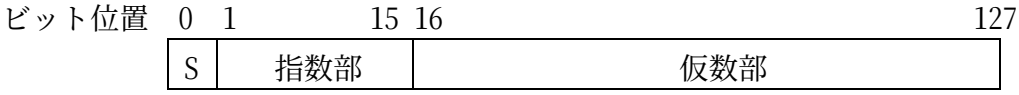
・実数型 4 バイト（単精度浮動小数点）の形式

SR8000 形式と同一です。

・実数型 8 バイト（倍精度浮動小数点）の形式

SR8000 形式と同一です。

・実数型 16 バイト（拡張精度浮動小数点）の形式



(2) データー互換オプション

M 形式又は SR2201(3050RX)形式で作成されたデーターを SR8000/MPP で入力したり、SR8000/MPP から M 形式又は SR2201（3050RX）形式で出力したりするには、ロードモジュールの実行時に RUNST オプションでデーター互換サブオプションを指定します。ただし、これらのデーターは、書式なし入出力文で入出力されたものでなければなりません。

データー互換サブオプションで変換の対象となるデーターの型と長さは、次のとおりです。

・実数型 4，8，16 バイト

・複素数型 8，16，32 バイト

ただし、複素数型のデーターは、実部と虚部がそれぞれ実数型とみなされて変換されます。

形式

RUNST(〔{CVIN|CVOUT}({3050R|M|SR8000}(n〔,n〕…))〕)

サブオプション

●CVIN｜CVOUT

データーの入出力時のどちらで変換するかを指定します。

CVIN

データーの入力時に変換します。

CVOUT

データーの出力時に変換します。

●3050R｜M｜SR8000

変換するデーターの形式を指定します。

・3050R

入力（CVIN）の場合、SR2201（3050RX）形式から SR8000 形式に変換します。
出力（CVOUT）の場合、SR8000 形式から SR2201（3050RX）形式に変換します。

・M
入力（CVIN）の場合、M 形式から SR8000 形式に変換します。
出力（CVOUT）の場合、SR8000 形式から M 形式に変換します。

・SR8000
変換しない。（デフォルト）

●n
データー変換の対象となる装置番号を指定します。

(3) 注意事項

①SR2201（3050RX）形式の単精度浮動小数点（実数型 4 バイト）及び倍精度浮動小数点（実数型 8 バイト）は、SR8000/MPP の浮動小数点形式と同一の形式です（形式は IEEE に準拠）

②SR8000/MPP の倍精度浮動小数点（実数型 8 バイト）及び拡張精度浮動小数点（実数型 16 バイト）を M 形式に変換する場合、オーバーフロー又アンダーフローが発生することがあります。オーバーフロー又はアンダーフロー発生時の動作は、FORTRAN 実行時オプション RUNST(CVCHECK)で制御できます。

③SR2201（3050RX）形式の拡張精度浮動小数点（実数型 16 バイト）を SR8000/MPP 形式に変換して使用すると、精度が低下します。

④M 形式の単精度浮動小数点（実数型 4 バイト）を SR8000/MPP 形式に変換する場合、オーバーフロー又はアンダーフローが発生することがあります。オーバーフロー又はアンダーフロー発生時の動作は、FORTRAN 実行時オプション RUNST(CVCHECK)で制御できます。

なお、本機能は環境変数 FTxxMyy によっても指定可能です。この環境変数の詳細については、SR8000/MPP FORTRAN 対応マニュアル「HI-UX/MPP 最適化 FORTRAN77 使用の手引き」及び「HI-UX/MPP 最適化 FORTRAN90 使用の手引き」を参照してください。

1.3 ファイル互換

(1) UNIX 系 FORTRAN 処理系からのファイル互換

SR8000/MPP FORTRAN では、書式なし入出力文に対応するファイル形式のデフォルト状態を、業界標準書式なしファイルとします。一方、SR2201 上の FORTRAN 処理系では、デフォルト状態は FORTRAN 固有ファイル（日立独自形式）としています。SR8000/MPP FORTRAN では、 SR2201 上の FORTRAN 処理系との書式無しファイルの互換性及び移行性を確保するため、 FORTRAN 固有ファイルについても入出力可能とし、既存ファイルの入力については、ファイル OPEN 時に FORTRAN 固有形式か業界標準書式なしファイルかを自動判別して処理します。

業界標準書式なしファイルと FORTRAN 固有ファイルとでは書式なし入出力時のファイ

ル形式が異なります。以下、その詳細について説明します。

業界標準書式なしファイルは、他社 UNIX 系 FORTRAN で作成した書式なしファイルと同様の形式であるため、他社 FORTRAN で作成した書式なしファイルを変換しないで使用できるという利点を持ちます。

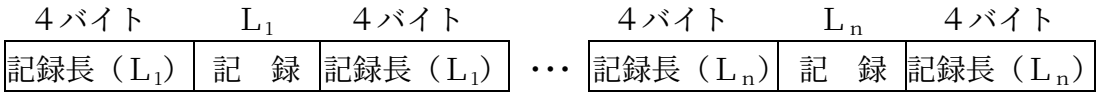
一方 FORTRAN 固有ファイルは、順番探査入出力と直接探査入出力のファイル形式が同一となります。このため、順番探査出力文で作成したファイルを直接探査入力文で入力したり、直接探査出力文で作成したファイルを順番探査入力文で入力したりできるという利点を持ちます。

①業界標準書式なしファイルの形式

業界標準書式なしファイルは、 SR2201 上の FORTRAN 処理系では、実行時オプション PORT(DSTDUF)（または PORT(STDUF)）指定時に使用できるファイル形式でした。しかし、SR8000/MPP FORTRAN では、デフォルトで業界標準書式なしファイルを使用可能とします。

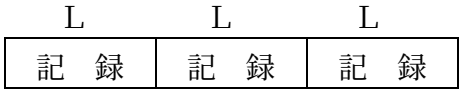
●業界標準書式無しファイル

・順番探査の場合



（凡例）L_1,…, L_n：並びごとの記録長（単位：バイト）

・直接探査の場合



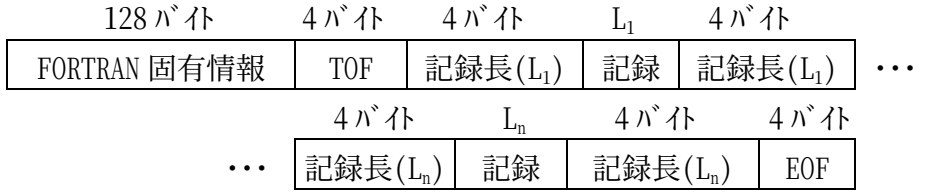
（凡例）L：OPEN 文の RECL 指定子で指定した長さ（単位：バイト）

②FORTRAN 固有ファイルの形式（日立独自形式）

FORTRAN 固有ファイルは、 SR2201 上の FORTRAN 処理系ではデフォルトの書式無しファイルの形式でした。しかし、SR8000/MPP FORTRAN では、業界標準書式なしファイルをデフォルトの書式無しファイルの形式とするため、FORTRAN 固有ファイルでの出力を行うには、実行時オプション PORT(FORTUF)を指定してプログラムを実行する必要があります。既存の FORTRAN 固有ファイルの入力については、自動判別機能によって、実行時オプションの指定なしで入力が可能です。

●FORTRAN 固有ファイルの形式

・順番探査、順番探査ともに同一の形式



（凡例）L_1, …, L_n：

順番探査の場合は並びごとの記録長（単位：バイト）

直接探査の場合はOPEN文のRECL指定子で指定した長さ（全て同一）
FORTRAN固有情報：FORTRAN固有形式を表現する情報
TOF，EOF：記録全体の前後に付加するFORTRAN固有形式の情報

1.4　64ビットアドレッシングモードのオブジェクトモジュールの作成

SR8000/MPP FORTRANでは、64ビットアドレッシングモードでオブジェクトモジュールを作成することができます。

(1) コンパイル時及びリンク時の指定方法

64 ビットアドレッシングモードのオブジェクトモジュールを作成する場合は、f77/f90コマンドオプションとして「-64」オプションを指定してください。

（例）

```
f77　a.f　-64
```

但し、f77/f90コマンドを用いて、64ビットアドレッシングモードのオブジェクトモジュールのリンクのみを行い、64ビットアドレッシングモードのロードモジュールを作成する場合にも、f77/f90コマンドオプションに「-64」オプションを指定する必要があります。

(2) -64オプション指定時の注意事項

-64オプションを指定した場合は、コンパイル時に次の点に注意してください。

①名前付き定数及び初期化項目には、2Gバイトを超えるデーターは使用できません。

②ASSIGN文、割り当て型GOTO文、及びFMT指定子で指定した変数は、整数型8バイトでなければなりません。-64 オプションは原始プログラム上で宣言された型のサイズを自動的に変更するものではないため、次に示す-intexp=full オプションを使用するか、必要があれば、これらの型変更をユーザー自身が行う必要があります。

③コンパイルオプション「-intexp=full」を指定することにより、整数型4バイトの変数・配列の型を一律に整数型8バイトの型に拡張することが可能です。しかし、このオプションは型を拡張した場合のプログラム挙動の正当性を保証するものではないため、指定時には実引数・仮引数の整合性、共通ブロックの要素の型の他プログラム単位との整合性等に対する注意が必要です。特にサービスサブルーチンの引数には整数型8バイトは使用できないため、-intexp=full オプションを使用する場合は-exsrvc オプションを合わせて指定する必要が有ります。この-exsrvc オプションを指定した場合、サービスサブルーチンの実引数として指定された整数型8バイト引数を自動的に整数型4バイト引数に変換します。但し、本オプション指定時にはサービスサブルーチンはユーザールーチンと置き換えが出来ません。

④-64オプション指定時には-hugeary オプションが仮定されるため、次の組込み関数は返却値の型がinteger*8に拡張されます。

LBOUND,SHAPE,SIZE,UBOUND,COUNT,MAXLOC,MINLOC

このため、integer*4 が返却値であることを前提としたプログラムでは、エラーとなる場合があります。

（例）

```
imax=max(k,size(array))
```

-64 オプションの場合、k は integer*4、size は integer*8 となるため、max 組込み関数の引数が一致しないでコンパイルエラーとなります。

なお、-intexp、-exsrvc オプションの詳細については、SR8000/MPP FORTRAN 対応マニュアル「HI-UX/MPP 最適化 FORTRAN77 使用の手引き」及び「HI-UX/MPP 最適化 FORTRAN90 使用の手引き」を参照してください。

(3) -64 オプション指定時のリンク注意事項

-64 オプションを指定した場合は、リンク時に次の点に注意してください。

①リンクするオブジェクトモジュールは、すべて「-64」オプションを指定してコンパイルされたオブジェクトモジュールでなければなりません。「-64」オプションを指定せずにコンパイルしたオブジェクトモジュールをリンクしようとした場合は、リンク時にエラーメッセージが出力されます。

(4) リモート DMA 関数・手続き使用時の注意事項

64 ビット版のプログラムにおいても、SR2201 でサポートしたリモート DMA 関数・手続きをそのまま使用することは可能ですが、以下の制約が発生します。

①2G バイトを越える配列位置に対する通信を行えません。

②64 ビット互換処理のためのオーバーヘッドが発生します。このため、64 ビット版のプログラムにおいては SR8000/MPP FORTRAN で新規に提供する 64 ビット版リモート DMA 関数・手続きを使用することを推奨します。なお、64 ビット版リモート DMA 関数・手続きとは従来の $R で始まる名称を $L で始まる名称として提供するものであり、インターフェイスとして一部引数の型を整数型 4 バイトから整数型 8 バイトに変更する必要があります。

（例）受信フィールド作成関数 $RCREATE について

32 ビットモード対応関数の場合

```
$RCREATE(iobjid, recvaddr, len, ifnum, irecopt, idesc)
  iobjid     ：整数型 4 バイト
  recvaddr   ：制限なし
  len        ：整数型 4 バイト
  ifnum      ：整数型 4 バイト
  irecopt    ：整数型 4 バイト
  idesc      ：整数型 4 バイト
```

64 ビットモード対応関数の場合

```
$LCREATE(iobjid, recvaddr, len, ifnum, irecopt, idesc)
  iobjid     ：整数型 8 バイト
```

recvaddr　：制限なし
len　　：整数型 8 バイト
ifnum　　：整数型 4 バイト
irecopt　：整数型 4 バイト
idesc　　：整数型 8 バイト

64 ビット版リモート DMA 関数・手続きの詳細については SR8000/MPP FORTRAN 対応のマニュアル「HI-UX/MPP リモート DMA 使用の手引き -FORTRAN-」を参照してください。

(5) ファイルサイズが 2G バイトを超える場合の注意

トータルのファイルサイズが 2G バイトを超えるファイルへの入出力は、32/64 ビットアドレッシングモードで共に可能です。ただし、非同期入出力文については、32 ビットアドレッシングモードで 2G バイトを超えるファイルへの入出力はできません。

また、各入出力のレコード長の制限は次のとおりです。

・順番探査入出力文

32 ビットアドレッシングモード：2G バイト未満

64 ビットアドレッシングモード：4G バイト未満

・直接探査入出力文

32/64 ビットアドレッシングモード：2097144k（2G-8k）バイト以下

1.5　ノード内並列化オブジェクトモジュールの作成

SR8000/MPP FORTRAN はノードを構成する各マイクロプロセッサーを並列に使用して DO ループを高速実行するオブジェクトモジュール（ノード内並列化オブジェクトモジュール）を生成する機能を持ちます。ここでは、ノード内並列化オブジェクトモジュールを作成する場合のコンパイルオプション指定方法及びロードモジュール作成時の注意事項を説明します。以下ノード内並列を単に「要素並列」と呼びます。

(1) コンパイル時及びリンク時のオプション指定方法

①要素並列化を行ったオブジェクトモジュールを作成する場合は、オプションとして「-parallel=n」オプションを f77/f90 コマンドに指定します。ただし、n には並列化レベルを指定します。並列化レベルの詳細は以下の通りです。

●並列化のレベル

0：並列化を行いません。

1：SECTION 型並列化のみ行います。

2：SECTION 型並列化と基本的な自動並列化を行います。

・変数・配列のプライベート化

・リダクション変数並列化

3：SECTION 型並列化とループ構造変換を伴う自動並列化を行います。並列化レベ

ル 2 に加えて以下の並列化処理を行います。
・ループ分配、分割、一重化
4：並列化レベル 3 に加えて以下の並列化処理を行います。
・パイプライン並列化
・条件下のインダクション変数並列化

なお、並列化オプション-parallel の詳細については、SR8000/MPP FORTRAN 対応マニュアル「HI-UX/MPP 最適化 FORTRAN77 使用の手引き」及び「HI-UX/MPP 最適化 FORTRAN90 使用の手引き」を参照してください。

（例）

```
f77  -04  -parallel=4  aaa.f  bbb.f
```

（注 1）複数の手続き／関数呼び出しを*POPTION の SECTION 型並列化指示により並列実行する場合、または、手続き/関数呼び出しを含む DO ループを*POPTION PARALLEL 指示により強制並列化する場合、並列に呼び出される手続き・関数自体をコンパイルする場合には以下の 2 点に注意してください。
・並列化が適用された部分がネストして実行されないようにするため、並列化オプション（-parallel オプション）を指定してはなりません。
・手続き／関数中で使用するループ制御変数等のローカル変数を各スレッド（並列処理単位）毎に独立に確保するため、-nosave オプションを明示的に指定します。
（例） f77 -nosave x.f
ローカル変数に対して、DATA 文によって初期値を指定している場合は、そのローカル変数は SAVE 属性となり、スレッド共有な変数となります。したがって、そのようなローカル変数を含むサブルーチンを並列実行部分から呼び出してはなりません。

（注 2）SECTION 型並列化を実施する場合の並列実行ブロックの数（SECTION オペランドの数）は、最大 8 個までしか指定できません。

②要素並列化を行うロードモジュールをオブジェクトファイルからのリンクだけで作成する場合は、オプションとして「-parallel」オプションを f77/f90 コマンドに指定してリンクします。
（例） f77 -parallel aaa.o bbb.o -o ccc

③要素並列実行部分の中から、リモート DMA 機能を使用することはできません。

(2) 要素並列化ロードモジュール作成時の注意事項

要素並列化ロードモジュールを作成する時は、次の点に注意してください。

①コンパイラーはデフォルト状態では、使用可能なプロセッサー数を動的に判断して要素並列処理を行うオブジェクトを生成します。

②リンクするオブジェクトモジュールの中に、-procnum=8 オプション（8 つのプロセッサーを占有した要素並列化を実施することを指定するオプション）でコンパイルされたオ

ブジェクトモジュールが一つでもあった場合、作成されるロードモジュールは、占有要素並列モード（ノード内のすべてのプロセッサーを要素並列実行のために使用できるモード）で実行可能なノードでだけ実行が許されます。ただし、ノードが占有要素並列モードとなるかどうかはシステム設定によって決定されます。

2. 行列計算副プログラムライブラリー

行列計算副プログラムライブラリーMATRIX/MPP、MATRIX/MPP/SSS に関する、SR2201 からの移行上の注意事項について説明します。

2.1 移行上の注意事項

(1) インターフェイス名称

SR2201 用の単一メモリー配置型インターフェイス及び分散メモリー配置型インターフェイスは、それぞれ SR8000/MPP 用の逐次処理用インターフェイス及び並列処理用インターフェイスに移行できます。

(2) 単一メモリー配置型インターフェイス移行時の注意点

SR2201 用の単一メモリー配置型インターフェイスでは、作業領域を行列計算副プログラムライブラリー内部で動的に確保していたため、引数中に作業領域を配列として確保する必要はありませんでした。SR8000/MPP 用の逐次処理用インターフェイスでは、従来の M シリーズや S シリーズで提供している行列計算副プログラムライブラリーと同様に、利用者作成プログラム中で作業領域を配列として確保し、引数として連絡する必要があります。ただし、引数の並び順に変更はないため、従来スカラー量として引数で連絡していた変数に対して、各副プログラムで必要な大きさの配列宣言を追加することで利用可能となっています。以下に実密行列の連立一次方程式を求解する機能 HDLGEM の例を示します。

①SR2201 用 MATRIX/MPP の HDLGEM を使用する場合

```
INTEGER N,NA,M,NB,IOPT,IP(N),IER
DOUBLE PRECISION A(NA,N),B(NB,M),EPS,WK          ← 作業領域 WK はスカラー量
：
CALL HDLGEM(A,N,NA,B,M,NB,EPS,IOPT,IP,WK,IER)
```

②SR8000/MPP 用 MATRIX/MPP の HDLGEM を使用する場合

```
INTEGER N,NA,M,NB,IOPT,IP(N),IER
DOUBLE PRECISION A(NA,N),B(NB,M),EPS,WK(N,2)     ← 作業領域 WK は配列
：
CALL HDLGEM(A,N,NA,B,M,NB,EPS,IOPT,IP,WK,IER)
```

(3) 単一メモリー配置型インターフェイスの実行形態

SR2201 用の単一メモリー配置型インターフェイスでは、1 ノードで実行する利用者作成

プログラムから呼び出した各機能の内部で自動的にノード間の並列処理をしていました。ノード間並列化に必要なデーターの分配・収集も機能内で行います。この自動並列化の為に、単一メモリー配置型インターフェイスを利用した利用者プログラムは、行列計算副プログラムライブラリーが提供する専用のコマンドでロードモジュールを起動する必要がありました。SR8000/MPP 用の逐次処理用インターフェイスでは、ノード間で並列実行せずに 1 ノードで実行し、ロードモジュールを起動するのに必要な専用のコマンドも不要となっています。

(4) SR8000/MPP 向きインターフェイスの提供

SR8000/MPP 用行列計算副プログラムライブラリーは、従来 M シリーズや S シリーズで提供しているインターフェイス及び SR2201 用のインターフェイスと互換性のあるインターフェイスを提供しています。これらの機能は、インターフェイス互換の制約の中で SR8000/MPP 向きに処理を最適化した機能です。SR8000/MPP 用の行列計算副プログラムライブラリーでは、インターフェイス互換の制約なしに、SR8000/MPP 向きに内部処理を変更し、最適化を強化した SR8000/MPP 向きのインターフェイスを新規に提供しています。SR8000/MPP 上では、重複するインターフェイスがある機能は、SR8000/MPP 向きのインターフェイスを優先的にご使用下さい。

2.2 SR8000/MPP 新規ライブラリー機能のご紹介

SR8000/MPP 用行列計算副プログラムライブラリーで新規に提供するライブラリー及び機能を以下にご紹介します。

(1) 要素並列適用ライブラリー

SR8000/MPPの特長であるノード内要素並列を適用したライブラリーを提供します。行列計算副プログラムライブラリーは、SR8000/MPPの性能を十分に引き出す為に、内部の処理方式をSR8000/MPP向きに書き換えて提供している製品です。その為、要素並列を適用した本ライブラリーをシステムが推奨するライブラリーとしており、本ライブラリーに対してSR2201と同一名称（MATRIX/MPP：libmatmpp.a、MATRIX/MPP/SSS：libmatmpps.a）を使用しています。なお、要素並列を適用しないライブラリーは、別ライブラリー名称（MATRIX/MPP：libmatmpp_sc.a、MATRIX/MPP/SSS：libmatmpps_sc.a）で提供しています。このライブラリーでは並列処理用インターフェイスを除く逐次処理用インターフェイスの全機能を提供しています。

各副プログラムのインターフェイスは、要素並列を適用したライブラリーと要素並列を適用していないライブラリーで同一です。リンク時に指定するライブラリーを変更することで、要素並列の適用・非適用を変更可能です。インターフェイスが共通の為に、要素並列を適用したライブラリーと適用していないライブラリーを混在して使用することは出来ません。

(2) 64bit アドレッシングモードライブラリー

SR8000/MPP では、64 ビットアドレッシングモードと、32 ビットアドレッシングモード

の 2 つのアドレッシングモードを提供します。行列計算副プログラムライブラリーも 2 つのアドレッシングモードに対応するライブラリーを各々提供しています。しかし、異なるアドレッシングモードのライブラリーを混在して使用することは出来ません。

(3) SR8000/MPP 用行列計算副プログラムライブラリー新規機能

SR8000/MPP 用行列計算副プログラムライブラリーでは、以下に示す新規機能（副プログラム）を提供します。

①SR8000/MPP 向きインターフェイス

SR8000/MPP 向きに内部処理を変更し、最適化を強化した SR8000/MPP 向きのインターフェイスを新規に提供しています。2.1(4)を参照下さい。

②スパースダイレクトソルバー

SR8000/MPP で新規に提供する MATRIX/MPP/SSS の機能で、実対称疎行列を係数行列とする連立一次方程式を求解します。スパースダイレクトソルバーでは、係数行列の非ゼロ要素を計算対象とすることで、従来から提供しているスカイラインソルバーよりもさらに演算に不要なゼロ要素を削減することが可能となり、行列の格納に必要なメモリーの所要量、計算量を削減して処理を高速化することが出来ます。三角分解の過程で fill-in 要素が発生しますが、前処理であるオーダリングにより fill-in 要素を削減することが可能です。

3. リモートDMA転送

リモートDMA転送は性能を引き出すために、ハードウェアアーキテクチャに依存したインターフェイスをもつ通信機能です。このため、SR2201のプログラムをSR8000/MPPに移行する際には、アーキテクチャの差異に伴う次の変更を行なう必要があります。また、SR8000では実装されていないストライド転送機能もSR2201と同様に使用できます。

なお、リモートDMA転送はノード間通信機能であり、ノード内通信には使用できません。ノード内通信にはMPIをご使用下さい。

3.1 移行におけるILP32、LP64プログラミングモデル共通の注意事項

SR2201ユーザーのプログラムは、ILP32プログラミングモデルで作成されたものです。SR8000/MPPは、ILP32プログラミングモデルとLP64プログラミングモデルの両方をサポートしており、SR2201からSR8000/MPPに移行する場合、両プログラミングモデルに共通した次の注意事項があります。

(1) 送信完了確認フラグ

combuf_send_with_flag関数、及びcombuf_make_tcw関数で指定する送信完了確認フラグの制約が変更となります。SR2201では先頭アドレスが32バイト境界、サイズが4バイトとなっていましたが、SR8000/MPPでは先頭アドレスが8バイト境界、サイズが8バイトに変更となります。先頭アドレスの境界については、32バイト境界から8バイト境界とSR2201に比べ、緩和される形となるので問題ありませんが、サイズについては4バイトか

ら8バイトに拡大されます。SR2201で送信完了確認フラグのすぐ後ろの4バイト領域を使用しているプログラムはSR8000/MPPでは誤動作してしまいますので注意してください。また、SR2201では、送信完了確認フラグは送信領域と異なるリモートDMA領域に設けることができましたが、SR8000/MPPでは、送信完了確認フラグは送信領域と異なるリモートDMA領域上に設けることはできません。なお、FORTRANのリモートDMA関数では、送信完了確認フラグを関数内部で隠蔽しているため、送信完了確認フラグのサイズ変更を意識する必要はありません。

(2) 受信完了確認フラグ

combuf_spin_wait関数、及びcombuf_spin_mwait関数で指定する受信完了確認フラグの制約が変更となります。SR2201では先頭アドレスが8バイト境界、サイズが4バイトとなっていましたが、SR8000/MPPでは、先頭アドレスが8バイト境界のままで、サイズが8バイトとなります。従って、SR2201で受信完了確認フラグのすぐ後ろの4バイト領域を使用しているプログラムはSR8000/MPPで誤動作しますので注意してください。なお、FORTRANのリモートDMA関数では、受信完了確認フラグを関数内部で隠蔽しているため、受信完了確認フラグのサイズ変更を意識する必要はありません。

(3) 受信完了確認の仕方

SR8000/MPPでは受信完了確認関数以外でデーターの受信完了を確認することはできません。受信データーの末尾、及び受信完了確認フラグを直接参照して受信完了を確認している場合、データーの到着が保証されません。また、複数の受信フィールドに対して逐次的にデーターを受信する場合、最後の受信フィールドへの受信確認ができたとしてもそれ以前のデーター受信は保証されません。なお、FORTRANのリモートDMA関数では、受信完了確認フラグを関数内部で隠蔽しているため、受信完了確認フラグの確認方法を意識する必要はありません。

(4) combuf_object_get関数

SR8000/MPPでは、combuf_object_get関数にCOMBUF_UNCACHEABLEオプションを指定した場合でも、このオプションは無視され、リモートDMAオブジェクト上のデーターはキャッシングされます。なお、FORTRANでは該当するオプションはサポートしておりません。このため、リモートDMAオブジェクト上のデーターは常にキャッシングされます。

(5) hmpp_barrier関数

SR2201では、hmpp_barrier関数のcolor引数に指定できる値は0～15ですが、SR8000/MPPでは0～7となります。これ以外の値が指定された場合、下位3ビットをcolor値として扱います。なお、FORTRANの場合は、$RBARRER（またはLCBARRIER）手続き及び$RFBARIR（またはLCBARRIER）の第1引数（color値）で、上記のhmpp_barrier関数と同様に下位3ビットをcolor値として扱います。

3.2 ILP32プログラミングモデルにおける注意事項

SR2201のプログラムをILP32プログラミングモデルのままSR8000/MPPに移行する場合、再コンパイルするだけで実行することができます。ただし、先に説明した注意事項が守られていることが前提となります。なお、combuf_object_get関数とcombuf_map関数で動的に確保できる領域の上限は1.5GBです。FORTRANの場合は、*coptionパラメーターで指定する割り付け名称サイズの上限が1.5GBとなります。

3.3 LP64プログラミングモデルにおける注意事項

SR2201のプログラムをSR8000/MPPでLP64プログラミングモデルに変更する場合、次の注意が必要です。なお、FORTRANについては「1.4 (4)リモートDMA関数・手続き使用時の注意事項」を参照下さい。

(1) 送信領域や受信領域などのサイズの型

4GBを超える物理メモリーを指定可能にするため、送信領域や受信領域などのサイズの型をunsigned int型からCb_size_t型に変更する必要があります。

(2) オブジェクトIDのCb_object_t型

オブジェクトIDのCb_object_t型がunsigned intからunsigned longに変更されます。リモートDMAオブジェクトIDについて、Cb_object_tで宣言している場合には問題ありませんが、unsigned int型で定義している場合、ソースコードの修正が必要です。

3.4 SR8000/MPPでの追加機能

(1) GET通信、アトミックメモリー操作機能

SR8000/MPPではGET通信、アトミックメモリー操作機能を新規にサポートします。GET通信とはデーター受信側から起動し、指定したリモートノード上のメモリー内容を直接読み込む通信です。また、アトミックメモリー操作とは、指定したリモートノード上のリモートDMA領域に対してメモリーの読み出しから書き込みまでを排他的に行うメモリー操作です。

4. MPI

4.1 SR8000/MPPでの追加機能

(1) MPI-2仕様のサポート

SR8000/MPP MPI機能では、MPIの仕様を拡張した「MPI-2:Extentions to the Message-Passing Interface, July 18 1997」仕様をサポートしました。このMPI-2仕様の特長は次の通りです。

①MPI-IO機能

複数のMPIプロセスから同一ファイルに同時アクセスするコレクティブI/Oインターフェイスや非同期I/O機能を持つMPI-IO機能をサポートしました。本機能により並列ファイルを高速にI/Oすることが可能になります。

②単方向通信サポート

従来のMPIにてサポートしている一対一通信は、送信側プロセスと受信側プロセスで

それぞれMPI送信関数、MPI受信関数を発行する必要があります。単方向通信機能は、windowと呼ぶメモリー領域をユーザープログラムが指定することによって、1プロセスからのみ通信関数を発行するだけで、メッセージ通信を実現する機能です。この機能には、共有メモリー操作（put操作、get操作）及び、リモート演算操作があります。

③動的プロセス生成機能

ユーザープログラムから動的にMPIプロセスを生成する動的プロセス生成機能をサポートしました。

④mpiexecコマンド

MPI-2仕様にて標準の起動コマンドとなるmpiexecコマンドをサポートしました。このmpiexecコマンドを使用することにより、MPMD起動（複数のMPIプログラムからなるMPIジョブを起動）することも可能となります。

(2) 64ビットアドレッシング用ライブラリーサポート

SR8000/MPP MPIではプロセッサーの64ビット化に対応し、SR2201 MPIでサポートしていた32ビットアドレッシング用ライブラリーとともに、64ビットアドレッシング用MPIライブラリーをサポートしました。

4.2 SR8000/MPPでの性能改善

(1) MPI通信の性能改善

SR8000/MPPでの高速な通信を利用し、SR2201 MPIでの通信よりも高速な通信性能（1.5GB/s）を実現しました。

(2) ノード内通信の性能改善

SR2201 MPIでは、同一ノード内のプロセス間通信をネットワークを介して通信を行なっていました。SR8000/MPP MPIでは、同一ノード内のプロセス間通信を行なう場合、ネットワークを介さずに通信するよう改善しました。これにより、同一ノード内のプロセス間通信では低レイテンシーで通信することができます。

(3) バリアー同期機能、ブロードキャスト機能のハードウェア転送機能サポート

従来のMPI_BARRIER関数及びMPI_BCAST関数では、一対一通信を組み合わせて実現しておりました。SR8000/MPP MPIのこれらのMPI関数は、ハードウェアバリア機能及びハードウェアブロードキャスト機能が使える条件である場合、このハードウェア機能を用いて動作するように改善しました。ハードウェアブロードキャスト機能が使える条件は、次の条件を全て満たす場合です。

①確保したノードの形状が矩形でかつ、ノードの属性がグローバル排他属性であること

②1ノードに1プロセスのみ起動すること

③コミュニケーターにMPI_COMM_WORLDを指定すること

5. OS コマンド／ライブラリー

SR2201 の OS では XPG3 準拠でしたが、SR8000/MPP の OS では XPG4 に準拠しています。ここでは OS の差異に伴うコマンド／ライブラリーの変更点を説明します。

(1) SR2201 の標準シェル（/usr/bin/sh）は Bourne-Shell ですが、SR8000/MPP の標準シェルは Korn-Shell です。

(2) SR8000/MPP の awk は nawk です。SR2201 互換の awk は/usr/bin/oldawk にあります。

(3) XPG4 化に伴い、一部の関数のプロトタイプ宣言が SR2201 と SR8000/MPP で異なります。プロトタイプ宣言の違いによってコンパイルエラーが起こる場合は、コンパイルオプションに-D_MPP_0203_SOURCE または-D_MPP_V2_SOURCE を指定してください。

(4) BUFSIZ 定数の値を 65536 に拡張していますので、short 型では扱えませんので注意してください。

(5) SR8000/MPP では国際化機能に対応しています。日本語 SJIS コードを使う場合は、LANG 環境変数に ja_JP.SJIS を設定してください。日本語 EUC コードを使用する場合は、ja_JP.eucJP を設定してください。

－以上－