



HPCを実現するソフトウェア Parallelnavi について

～プログラム開発環境を中心に～

富士通株式会社
2000年9月8日

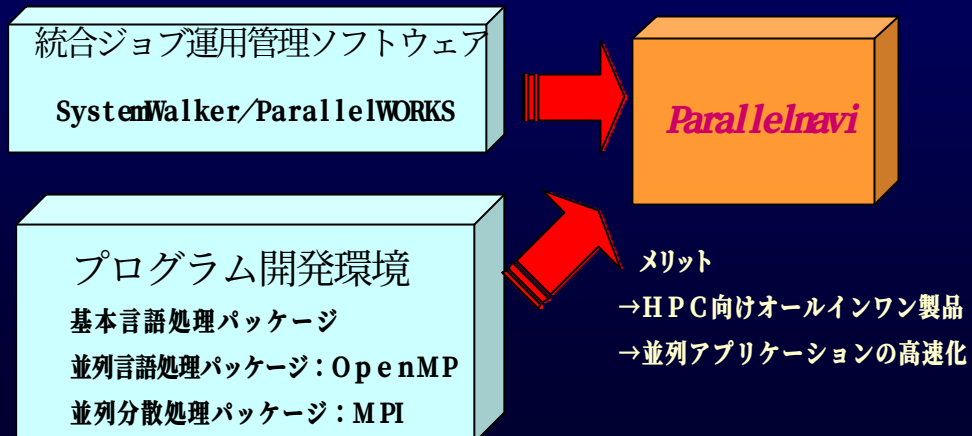
All Rights Reserved, Copyright © FUJITSU LIMITED 2000

スカラSMPマシン上でHPCを実現するソフトウェア、Parallelnaviについてプログラム開発環境を中心に報告する。

Parallelnaviとは？

現行製品は単品で提供

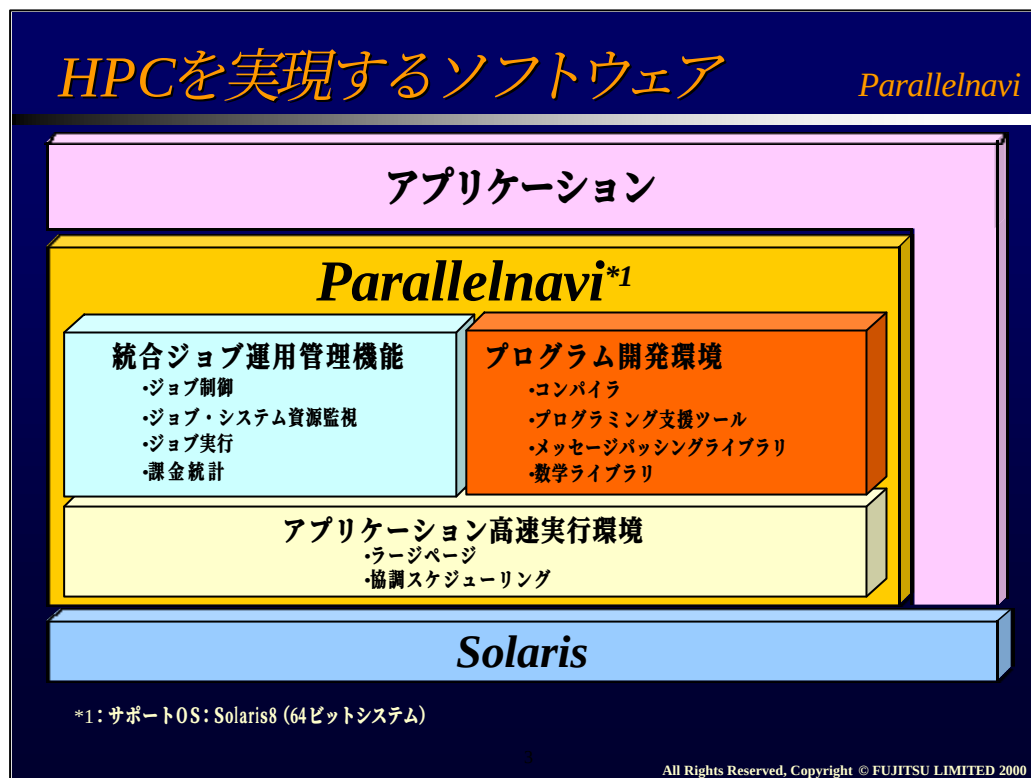
HPC向け統合製品



All Rights Reserved, Copyright © FUJITSU LIMITED 2000

Parallelnaviとは、単品で製品提供していた統合ジョブ運用管理ソフトウェアであるSystemWalker/ParallelWORKSや、各種言語処理パッケージをまとめた、HPC向け統合製品である。

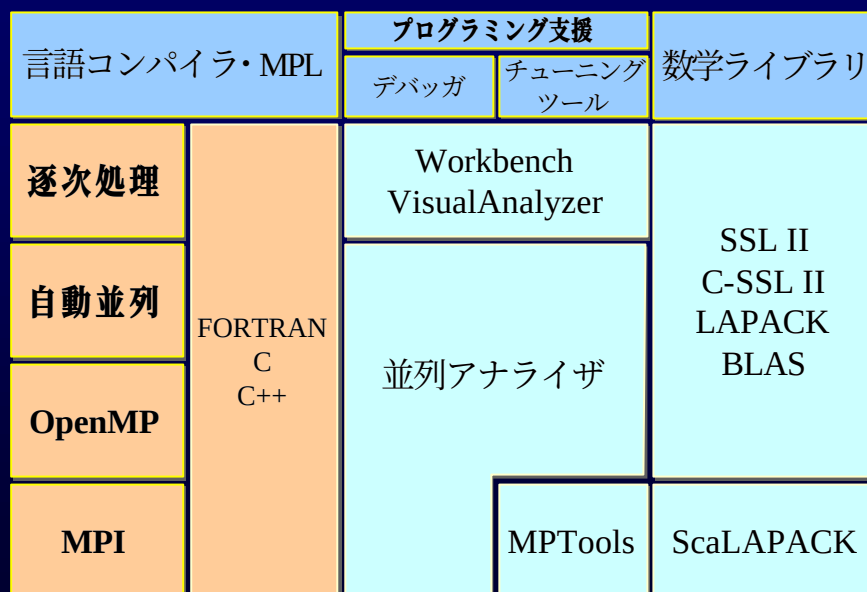
メリットは、「オールインワン製品」であることと、「並列アプリケーションの高速化」である。



HPCを実現するソフトウェアParallelnaviの体系図である。Parallelnaviには、「統合ジョブ運用管理機能」「アプリケーション高速実行環境」及びこれから説明する「プログラム開発環境」で構成されている。

プログラム開発環境

Parallelnavi



All Rights Reserved, Copyright © FUJITSU LIMITED 2000

プログラム開発支援環境について、コンパイラおよびツール、数学ライブラリの位置付けを図示した。

今回特に、C,C++については、自動並列及び、OpenMPを新たにサポートしている。

標準性の追求

Parallelnavi

- 最新の国際規格及び業界標準規格の採用

Fortran	ISO/IEC 1539-1:1997, JIS X3001-1:1998(Fortran95)
C	ISO/IEC 9899:1999(C99規格), X3.159-1989(ANSI C規格), K&R仕様
C++	ISO/IEC 14882:1998
OpenMP	OpenMP Fortran Application Interface Version1.1, OpenMP C and C++ Application Interface Version1.0
MPI	MPI2.0

All Rights Reserved, Copyright © FUJITSU LIMITED 2000

Parallelnaviのプログラム開発環境の狙いの1つに「標準性の追求」がある。Fortranは通称Fortran95規格、C言語はC99規格等、各言語は最新の国際規格及び業界標準規格を採用している。

高速性の追求

Parallelnavi

- 高度な最適化コンパイラ

SPARC 向け	データプリフェッチ, 命令スケジューリング, etc
SPARC64 GP 向け	M&A 命令生成, etc

- 並列化コンパイラ

自動並列化	コンパイルオプションによる容易なループ自動並列化 指示行挿入による柔軟な並列化機能
OpenMP	指示行挿入による, より高度で柔軟な並列化機能

- 高速数学ライブラリ

SPARC64 GP 向けに高度にチューニング
並列処理(OpenMP, MPI)による解法時間短縮

All Rights Reserved, Copyright © FUJITSU LIMITED 2000

Parallelnaviのプログラム開発環境の狙いのもう1つが「高速性の追求」である。

1CPU向けには、汎用SPARCチップ向けの高度な最適化やSPARC64 GPチップに特化した最適化を行っている。

並列化コンパイラとして、自動並列化及びOpenMPをサポートしている。

また、高速な数学ライブラリも用意している。

1CPU向け最適化の効果

Parallelnavi

現行製品 (Fortran&C V4. 0. 2) とParallelnaviとの
SPEC BMTのSPEC Ratio比較

現行製品 *Parallelnavi*

SPECfp2000	1.0	1.34倍
SPECint2000	1.0	1.16倍

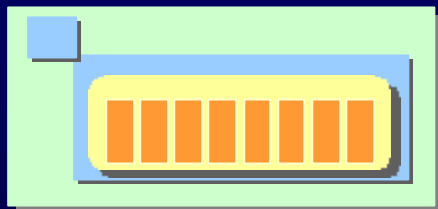
All Rights Reserved, Copyright © FUJITSU LIMITED 2000

1CPUの性能については、現行製品とParallelnavi製品のSPEC2000のpeak性能を同じハード上で比較すると、SPECfp2000で1.34倍、SPECint2000で1.16倍、に性能が向上している。

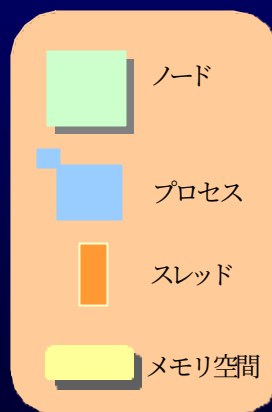
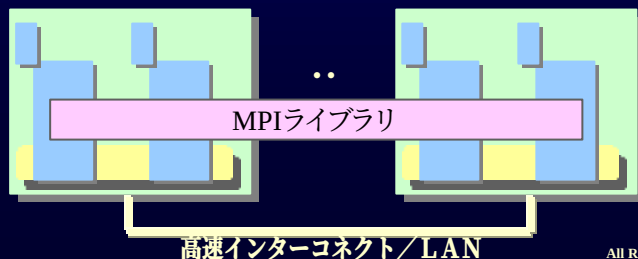
並列処理の実現

Parallelnavi

- 自動並列／OpenMP(ノード内並列処理)



- MPI(ノード内・ノード間並列処理)



All Rights Reserved, Copyright © FUJITSU LIMITED 2000

ここでは、SMPでの並列処理の実現について説明する。

ノード内並列処理として、コンパイラによる自動並列、またはOpenMPによる並列処理がある。この場合コンパイラは、スレッドと呼ぶ並列処理単位を生成するオブジェクトコードを生成する。プログラム実行時には、一つのプロセス配下で複数のスレッドが同時（並列）実行される。

一つのスレッドが通常1CPU下で実行される。並列度（スレッド数＝動作CPU数）は実行時の環境変数として定義される。

このコンパイラによる自動並列化/OpenMPは小～中規模問題を容易に並列処理したい場合に用いられる手法であり、従来のベクトル機のコンパイラを利用した自動ベクトル化に相当するものである。

問題規模が大きくなった場合には、自動並列/OpenMPとMPIを組み合わせたプログラミングモデルが考えられる。これはコンパイラにより生成されたスレッド並列を行うプロセス間で、さらにMPI通信を用いて並列処理を行う手法であり、単一OS(一台のSMPマシン)内でも、あるいは問題規模がさらに大きい場合には複数のOS(複数台のSMPマシン)をまたがってMPI通信を利用できる。本手法はベクトル並列機（VPP）におけるベクトル並列処理に相当するものである。

- 自動並列化機能
 - Fortran/C/C++コンパイラで利用可能
 - コンパイルオプション、指示行による容易な並列化
- OpenMP
 - Fortran/C/C++コンパイラで利用可能
 - OpenMP Fortran規格 1.1 仕様及びOpenMP C/C++ 規格 1.0仕様に準拠
 - 豊富な機能
 - ループ並列、タスク並列、データスコープの指定、リダクション演算の指定、同期/排他制御など
- ツール（並列アナライザ）
 - マネージャ: GUI統合開発環境(翻訳、編集、実行支援、ツールの起動)
 - デバッガ : 並列プログラム(OpenMP, MPI)のデバッグ
 - サンプラ : 手続き/ループ/LWP単位および1行ごとのコスト解析
- 数学ライブラリ
 - OpenMP並列版SSL II
 - SSL II, C-SSL II, BLAS, LAPACK(逐次スレッドセーフ版)

All Rights Reserved, Copyright © FUJITSU LIMITED 2000

自動並列及びOpenMP（スレッド並列）の開発環境について説明する。

自動並列化については、Fortran/C/C++コンパイラで利用可能であり、コンパイラオプション又は指示行により容易な並列化が実現できる。

OpenMPについても、Fortran/C/C++コンパイラで利用可能であり、豊富な並列化の記述により高速化が実現可能である。

ツールについては、並列アナライザによりデバッグ及びチューニングが可能

である。

数学ライブラリについては、スレッド並列で動作するOpenMP版SSLII及びスレッドセーフ版の各種数学ライブラリが用意されている。

高度な自動並列化

Parallelnavi

自動ベクトル化技術を応用した、高度な自動並列化

例) LINPACK BMTコードの高コスト部

→複雑な引数を持つ手続き引用を含むループも自動並列

```
p      do 30 j = kp1, n
p      t = a(l,j)
p      if (l .eq. k) go to 20
p      a(l,j) = a(k,j)
p      a(k,j) = t
p      20  continue
pi     call daxpy(n-k,t,a(k+1,k),1,a(k+1,j),1)
p      30  continue
```

300元4CPU並列でほぼ4倍の並列効果

10

All Rights Reserved, Copyright © FUJITSU LIMITED 2000

Parallelnaviでは、VPPで長年培った自動ベクトル化技術を応用した、高度な自動並列化を実現している。例えば、LINPACK BMTコードの高コスト部分は、複雑な引数を持つ手続き引用が存在する。この部分をもコンパイラオプションの指定だけで自動的に並列化を行うことができる。

ループ並列

```
:  
DO I=1,1000  
  B(I)=(A(I)+A(I+1)/2.0)  
END DO  
:
```



```
:  
!$OMP PARALLEL DO SCHEDULE(STATIC)  
DO I=1,1000  
  B(I)=(A(I)+A(I+1)/2.0)  
END DO  
!$OMP END PARALLEL DO  
:
```

CPU1

```
:  
!$OMP PARALLEL DO SCHEDULE(STATIC)  
DO I=1,500  
  B(I)=(A(I)+A(I+1)/2.0)  
END DO  
!$OMP END PARALLEL DO  
:
```

CPU2

```
:  
!$OMP PARALLEL DO SCHEDULE(STATIC)  
DO I=501,1000  
  B(I)=(A(I)+A(I+1)/2.0)  
END DO  
!$OMP END PARALLEL DO  
:
```

OpenMPのプログラミングの1例である。2つの簡単な指示行を挿入するだけで容易に並列プログラミングが可能となる。

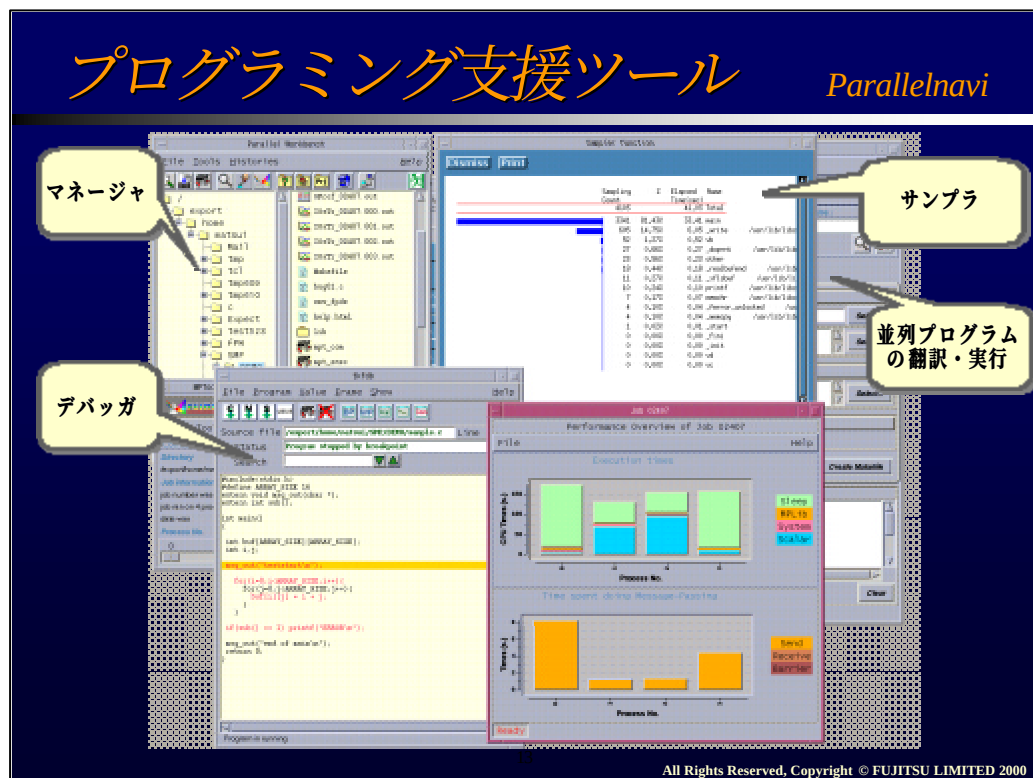
- メッセージパッシングライブラリ (MPI)
 - 最新のMPI規格(MPI2.0)に準拠
 - SMPノード内 (共有メモリ通信) , ノード間 (TCP/IP通信) 双方に対応した並列プロセス間通信
 - サポート機能
 - 1対1通信, 集団通信, 派生データ型, コミュニケータ, プロセス・トポロジー, 動的プロセス生成/管理, 片側通信, 並列入出力など
- ツール (MPTools)
 - MPIのプログラム解析支援
 - 各プロセスの情報, ジョブの収集した解析情報の集計表示など
 - GUIによる容易な操作性
- 並列線形計算ライブラリ (ScaLAPACK)
 - 逐次数学ライブラリLAPACKのMPI対応版
 - 提供ライブラリ
 - 線型方程式, 最小二乗問題, 固有値問題, 特異値分解など

MPI(プロセス並列)の開発環境について説明する。

MPIは最新のMPI規格2.0に準拠し、SMPノード内やSMPノード間双方に対応した各種並列プロセス間通信をサポートしている。

ツールについては、MPToolsによりMPIプログラムの解析支援が可能である。

数学ライブラリについては、ScaLAPACKライブラリを用意している。



プログラミング支援ツールの画面例である。

マネージャ、デバッガ、サンブラ及び並列プログラムの翻訳・実行等をGUIベースで動作させることができる。

OS機能との連携

Parallelnavi

- 協調スケジューリング機能
 - 複数並列ジョブ実行時のスレッドの同時スケジューリング
➡ 同期待ち時間の短縮化, 並列ジョブ実行時間の短縮化
- ラージページ機能
 - ページサイズ拡張によるアドレス変換キャッシュ(TLB)のミスヒット削減
 - ページのロックによるページングの抑制
➡ メモリサイズが大きいプログラムの実行性能を保証

【効果の例】SPECfp2000

プログラム	メモリサイズ MB	効果
171.swim	190MB	47%UP
178.galgel	148MB	142%UP
301.apsi	191MB	33%UP

14

All Rights Reserved, Copyright © FUJITSU LIMITED 2000

Parallelnavi製品では、OSと連携して、JOBの高速実行を可能にする。

協調スケジューリング機能は、複数の並列ジョブ混在時に、同一JOBを実行しているタイムスライスと同時に実行できるように制御することにより、同期

ズレによる性能低下を防止する。

ラージページ機能は、ページを大きくすることにより、TLBミスを減少させると共に、ページを固定することにより、大規模プログラムの性能低下を防ぐ

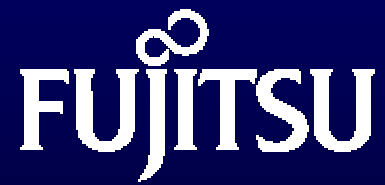
ことができる。

試しに、現製品に対してラージページを使用すると、SPECfp2000において、

このような大きな効果があった。

PRIMEPOWERにフォーカスした プログラム開発環境

PRIMEPOWERにフォーカスしたプログラム開発環境をParallelnaviで提供している。今後も継続的にエンハンスに取り組んで行く。

The Fujitsu logo is centered in the upper half of a dark blue rectangular box. It features the word "FUJITSU" in a white, serif, all-caps font. An infinity symbol (∞) is positioned above the letter "j", with its top loop resting on the letter "F" and its bottom loop resting on the letter "i".

FUJITSU

THE POSSIBILITIES ARE INFINITE