17

数値風洞のオペレーティングシステム

福田正大* 末松和代* 土屋雅子*
大空瞭** 工内隆** 坂本喜則**

The Operating System for Numerical Wind Tunnel

by

Masahiro FUKUDA, Kazuyo SUEMATSU and Masako TUTIYA
National Aerospace Laboratory
Akira OZORA, Takashi KUNAI and Yoshinori SAKAMOTO
Fujitsu Limited

ABSTRACT

Numerical Wind Tunnel (NWT) is a CFD-oriented parallel computer system with distributed memory.

Conventional Operating System (OS) has been developed mainly based on computers with shared memory. Each processor of NWT has its own individual OS and each OS needs to work in collaboration with each other. Therefore OS itself has a characteristic of programs processed in parallel.

In this paper, OS functions required for NWT is discussed with the parallelism of OS taken into account to attain effective managements of hardware resources and high-speed processing of CFD programs.

1. はじめに

数値風洞は，CFD (Computational Fluid Dynamics) プログラムに特化した高い並列化効率（実行効率）の実現を目的とし，ベクトル計算機を要素とする分散メモリ型並列計算機システムである[1)]。

分散メモリ型並列計算機システムでは，各プロセッサに1つのオペレーティングシステム (OS) が存在する。数値風洞のOSでは，各プロセッサ上のOSが協調しながら，複数プロセッサ上で同時実行されているCFDプログラム（並列処理プログラム）を高速に実行する必要がある。従来のOSは，共用メモリ型計算機システムを前提としており，分散メモリ型でのOSの問題点は明確になっていない。本論文では，分散メモリ型でのOSの問題点を明確にしながら，数値風洞におけるOS要件について，運用機能を中心に考察している。

2. 数値風洞のシステム構成

(1) 数値風洞のハードウェア

数値風洞のハードウェアは，以下の装置から構成されており，高速なシステム記憶 (SSU) 又はLAN (Local Area Network) を介して，フロントエンドプロセッサ(FEP)であるFujitsu-VP2000シリーズと接続されている。

(a) 演算処理要素計算機 (PE)

新規アーキテクチャのプロセッサであり，ベクトル演算機能を持つが，クロスバネットワーク以外の入出力装置は接続されていない。数値風洞には最大200台程度のPEが接続可能である。

* 航空宇宙技術研究所
** 富士通株式会社

This document is provided by JAXA.

(b) コントロールプロセッサ(CP)

数値風洞の制御用プロセッサであり，SSU/LANによりFEPと接続されている。PEと同一のスカラユニットを持つが，ベクトル演算機能は付いていない。なお，数値風洞には複数台のCPが接続可能である。

(c) 結合ネットワーク(クロスバネットワーク)

CP間，CP-PE間，PE間を接続する高速なデータ通信路であり，PE間の距離を等距離とするため，クロスバネットワークを用いている。同時にハードウェアとして同期機能を持っている。

(2) 数値風洞の使用方法と各OSの機能分担

数値風洞は，CFDプログラムの実行専用のバックエンドプロセッサ(BEP)として使用する。すなわち，FEP-OSからCP-OSへジョブ実行が依頼され，CP-OSがジョブ実行時に必要な台数のPEを割り当てた後，対象となったPE-OSにジョブ実行を依頼する。

PE-OSは，ジョブの実行状態を監視し，ジョブ実行が終了すると，PE-OSがCP-OSにジョブ終了を通知し，さらに，CP-OSがFEP-OSにジョブ終了を通知する。

したがって，FEP，CP及びPE上の各OSは，主に以下の機能を分担することになる。

(a) FEP-OS

(i) 数値風洞の起動/停止

(ii) ジョブの実行依頼

(iii) ジョブ終了処理

(b) CP-OS

(i) PE管理(PE割当/解放等)

(ii) ジョブスケジューリング

(iii) ファイル管理

(c) PE-OS

(i) ジョブの実行/監視

(ii) ジョブ終了処理

ジョブ実行に必要なファイル(プログラム，データ)は，基本的にFEP配下の外部記憶装置に存在し，必要な時点で各PEからSSU/LAN及びクロスバネットワーク経由でデータ参照が行われる。演算結果は，これとは逆の順序でFEP上に書き出される。

(3) OSアーキテクチャ

数値風洞(CP，PE)は，新規アーキテクチャのプロセッサにより構成される分散メモリ型並列計算機システムであるため，OSには，分散処理向きであり，高い移植性とオープン性を合わせ持つUNIX(注1)をベースに考える。

注1) UNIXオペレーティングシステムはUNIX System Laboratories Inc.が開発し，ライセンスしている。

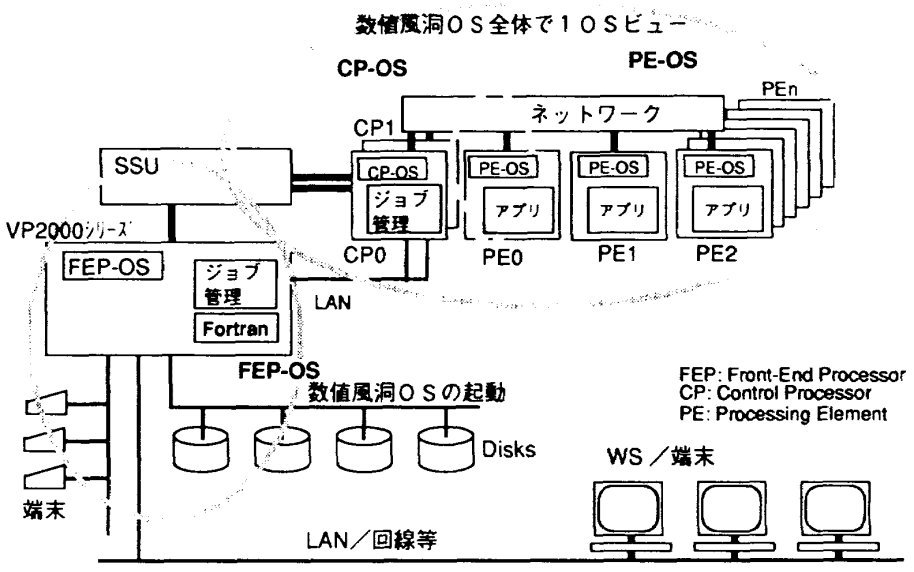

図1 数値風洞OSのシステム構成

This document is provided by JAXA.

3. OSの基本要件

数値風洞のOSには，以下に示す3つの基本要件が存在する。

(1) CFDプログラムの高速実行

CFDプログラム（並列処理アプリケーションプログラム）を高速に実行することが，第一に達成すべき課題である。このためには，並列処理アプリケーションプログラム（以下，並列処理プログラムと記述する）への支援機能として以下に示す機能[2]が必要である。

(a) 並列処理プログラムの高速生成（多重fork）機能

(b) 並列処理プログラム内のプロセス間高速通信・同期機能

(c) 並列処理プログラム内で仮想的に共用されるメモリ空間（グローバル空間）機能

(d) 高速入出力処理機能

(e) 性能情報（チューンナップ情報）の取得/出力機能

(2) 分散メモリ型のOSインタフェース

従来OSでは，共用メモリ型での並列処理や分散メモリ型の並列処理としてリモートプロシジャコールまでしか実現されていない。一方，数値風洞では，PEのOS上に分散配置された各並列処理プログラムを扱うため，OS自身が並列処理プログラムとして動作する機能を実現しなければならない。

また，既存プログラム資産の保証と並列化を推進する事が重要であるが，数値風洞においても，従来OSと同一のAPI (Application Program Interface) を実現することが望ましい。

このため，以下に示す機能が必要である。

(a) OS間の高速通信・同期によるシステム全体の効率的な資源管理

(b) 従来APIの保証（分散メモリ型への拡張）

(3) 運用機能上の要件

数値風洞を運用していく上で，従来運用を継承しながら，数値風洞を効率的に使用できる機能が必要である。このため，現運用機能に加え，以下の機能が必要である。

(a) 効率的なジョブスケジューリング機能（ジョブの受け渡し，優先度制御，ジョブ監視等）

(b) 並列資源制御機能（PE割当/メモリ割当）

(c) PE間のOS通信・同期機能

(d) FEP−数値風洞間の高速なファイル共用/保全性保証機能

(e) 性能情報（チューナブルパラメータ等）の出力機能

(f) 動的PE管理機能（縮退運転，スワップ機能等）

(g) RAS (Reliability, Availability and Serviceability) 機能（異常PE切離し/再組込み，多重化による耐故障性等）

(h) 障害解析機能（ダンプ取得，事象トレース，エラー情報ロギング機能等）

(i) 複数PE間の高精度なOS時刻設定機能（同時性の保証）

(j) ベクトル機能支援機能（ベクトル演算のためのメモリ割当やベクトル用領域サイズの動的変更等）

以下に，いくつかの項目について概要を述べる。

4. 並列処理プログラム支援機能

並列処理を行うアプリケーションプログラムの支援機能として，プロセスの親子関係をシンプルに保ちながら多数の子プロセスを高速に生成したり，並列処理プログラムのプロセス間の高速なデータ転送/送信，PE間同期等の処理実行を支援する機能である。これらの支援機能を以下に整理する。

(1) PE間の多重fork機能

シンプルな親子関係を保ちながら，多数の子（並列）プロセスをPE間に渡って高速生成する機能が必要である。この機能は，子プロセスの高速生成だけではなく，並列処理プログラムの設計を容易にする効果もある。この機能は2進木方式のfork機能によって実現する。

(2) PE間の高速通信・同期

多数の並列処理プログラム内のプロセス間で高速にデータ通信する機能と，並列プログラム内で指定されたプロセス群の間で同期を取る機能が必

This document is provided by JAXA.

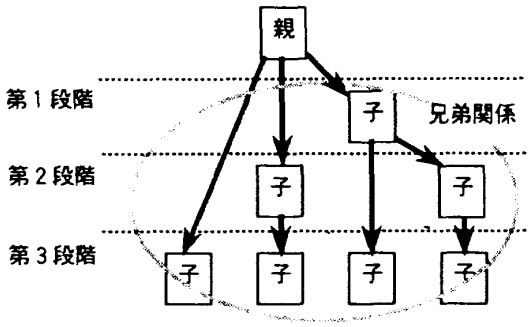

図2 PE間の多重fork機能

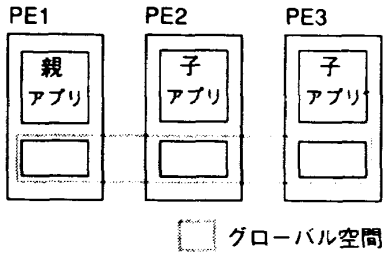

図3 PE間の仮想共用メモリ
（グローバル空間）

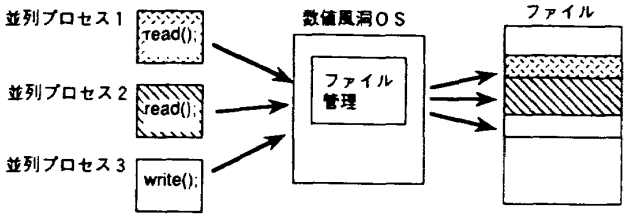

図4 read/write要求単位のアトミック化

要である。

(3) PE間の仮想共用メモリ

並列処理プログラムでは，巨大配列データ等を出来るだけ最適に各PEに分割配置するが，他PEに配置したデータが必要になる場合がある。このため，各PEに最適に分散配置しながら，各PE間で相互参照できる仮想的な共用メモリ空間（グローバル空間）が必要である。グローバル空間により，従来と同様なFortranプログラムが並列処理に適用可能となる。

このとき，並列処理プログラムからOSを経由せずに直接共用できることも，性能上必要である。

(4) 複数PEからの同一ファイルの高速参照

複数PE上で実行中の並列処理プログラムから同一ファイルを参照する場合がある（通常は同一並列処理プログラム内のプロセスから同時発行されることが多い）。このため，ファイル又はファイルシステムを複数PE間で共用しながら，かつ高速に参照する機能が必要である。

ベクトルデータは，繰返し参照が少ないという性格から，キャッシュレスかつ連続アクセスとすることにより，キャッシュ操作のオーバヘッドを削減することが可能である。また，read/writeの要求単位で処理のアトミック化を行う必要がある。

(5) 性能/チューニング情報の取得/出力機能

並列処理プログラムは，まだ技術が確立していないため，性能向上のためのチューニング情報が重要である。このため，性能情報の取得/出力機能や，各種の基準値をシステムパラメタやコマンド等により動的に変更できる機能が必要である。

(6) 並列プロセス終了時の並列資源の解放

ジョブ終了時において，並列処理プログラムの資源（クロスバネットワーク上の通信データも含む）を解放する必要がある。

5. 分散並列処理OS

分散並列処理OSは，OS自身が並列処理プログラムであるため，OSの制御表を含め，OS資源の最適な分割配置を実現する必要がある。また，分散メモリ型の並列計算機システムでは，ボトルネックを発生させないために出来るだけ各PE内でOS処理を完結させ，他PE/CPのOSとの排他制御を不要にする必要がある。

このため，以下の2つの観点を考慮してOS処理を実現する必要がある。

(1) OS処理の階層化

OS処理を，各プロセッサ(PE/CP)内に閉じる処理，ジョブ（並列処理プログラム）内に閉じる処理，及びシステム全体で制御すべき処理の3階層に分類する。これにより，OS処理の実現方式が明確になる。以下に各階層の例を示す。

(a) 各プロセッサ内の処理

(i) ジョブ実行/ジョブ監視

(ii) メモリ管理/ベクトル機能

(iii) プロセス（プロセッサ）内の優先度制御

(iv) 異常終了処理（割り込みハンドリング）

(b) ジョブ（並列処理プログラム）に依存する処理

This document is provided by JAXA.

(i) グローバル空間
(ii) ファイル及びファイルシステム（共用）
(iii) ジョブ内プロセス間の通信/同期（排他制御）

(c) システム全体で制御すべき処理
(i) ファイルシステム管理/高速入出力
(ii) ハードウェア資源の有効かつ効率的な管理
(iii) ジョブの優先度制御
(iv) 複数ジョブ間の通信/同期（セマフォ・メッセージ機能による排他制御）

(2) OS制御方式

(1)の(a), (b)については，OS間で排他制御が必要となるが，この場合，以下の2つの方式がある。OS処理の性格により性能が異なるため，各処理ごとの固有の設計が必要となる。

(a) 分散化方式

分散化方式は，分散メモリ型における性能上のボトルネックを起こしにくくするが，その反面OS間の排他制御は複雑になり，集中化方式よりもOS間デッドロックの原因になる可能性が強い。本方式も排他制御が頻繁に起こる場合は，集中化方式よりも性能が落ちることがある。

(b) 集中化方式

OS設計は分散化方式よりも容易だが，プロセッサ数が多数の分散メモリ型システムでは，性能上ボトルネックの原因となりやすい。

以上のことから，分散メモリ型のOSは，各処理の性格に合わせて分散化と集中化を併存させて管理していく必要がある。また，各システムによっても性能上採用すべき方式が異なるため，経験を積みながら順次機能をアップしていく必要がある。分散並列処理OSは，ネットワーク通信等，タイミング依存/リカーシブなエラーのリカバリ処理等，設計に著しく困難な問題を含んでいる。

6. ジョブスケジューリング機能

数値風洞のOSは，FEPから依頼されたジョブを受け付け，ジョブの優先度に基づいて資源管理を行いながらジョブの実行を制御する。このため，PEの使用状況管理，PE内の資源管理の2段階の管理機能がOSに必要である。

(1) 基本的なジョブスケジューリング機能

FEPと数値風洞とのジョブ要求の受渡しには，UNIXネットワーク間のバッチ機能として国際的な標準となっているNQS（注2）機能をベースに考える。標準NQS機能を以下に示す。

(a) システム内，ジョブクラス内の多重度制御機能（同時実行ジョブ数の管理）
(b) ジョブ投入時/受渡し時の正当性検査機能（投入権，設定値の検査）
(c) ジョブ/プロセス毎の資源制御機能（CPU/メモリ等の使用量制限）

これに対し，数値風洞では，並列アプリケーション管理のために以下の追加機能が必要である。

(a) ジョブクラス，ジョブ毎の並列資源管理（属性）
(i) 並列度
並列処理の実行に必要なPE台数
(ii) 資源制限
経過時間，CPU時間，グローバル空間を含めた最大メモリ使用量

(2) PE割当管理機能

PE割当管理機能は，並列ジョブ要求の実行に必要なPE台数（並列度）を割り当てる機能であり，PMS (Partition Management System) 機能と呼ぶ。PMS機能は，システム内の各PEの状態（使用可能PE，ジョブ実行中PE，オフラインPE等）を管理する。

並列ジョブの実行用PEの割当方式には，以下の2つが考えられる。数値風洞は高速並列処理を目的とするため，下記のジョブの即時実行優先方式が効率良く，1－並列プロセスが1台のPEを専有することにより最大性能を発揮できる。

(a) ジョブの即時実行優先方式（CFD特化型）
(i) 1－並列プロセス/1PE（ただし，システム全体では多重ジョブ）

(b) PEの効率的管理優先方式
(i) 空きPE数の最小化

注2) NQS (Network Queuing System) は，アメリカ航空宇宙局 (NASA) の依頼により開発された。

This document is provided by JAXA.

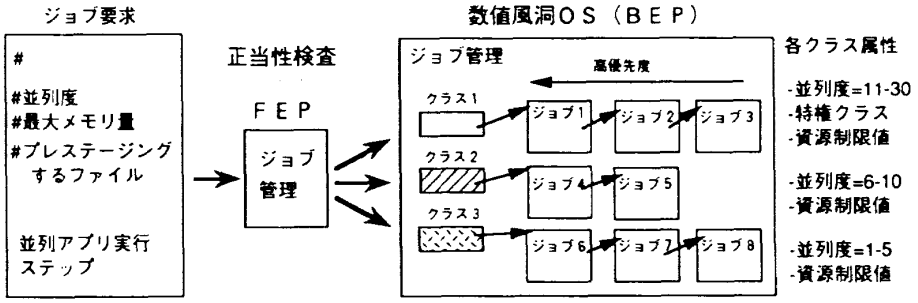

図5　ジョブスケジューリング機能

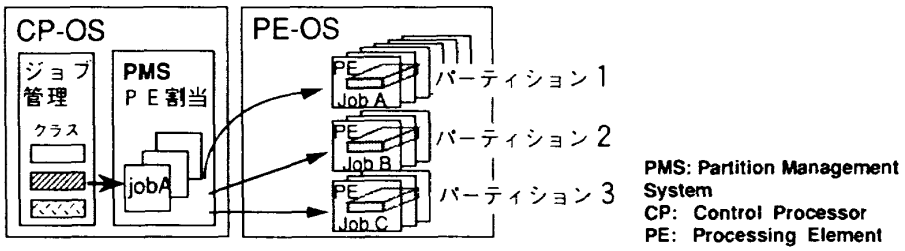

図6　PMS 機能

(II)　ジョブクラス毎の最大利用 PE 台数を制御

(3)　ジョブ優先度制御

実行中のジョブを中断 / 停止してでも，高優先度ジョブを実行する機能を考えておく必要がある。

7.　データ管理機能

数値風洞内でジョブ実行のために必要となったファイル（プログラム，データ）及び実行時に出力されたファイルをFEPとの間で高速に転送するため，以下に示す効率的なSSUの管理機能やデータ転送機能等が必要である。

(1)　SSU常駐化ボリューム機能

小容量だが参照頻度の高いファイルを保持するために，SSU 上に磁気ディスクと同等な常駐化ボリュームを構築する機能を設ける。

(2)　SSU キャッシュ機能

SSUに入らない程の大容量ファイルや参照頻度の低いファイルを保持するために，SSU をキャッシュとしてファイルを保持するデータ階層化機能を設ける。ここでは，必要な時点でFEP配下のディスクからSSUへファイルデータが転送される。また，参照頻度が低かったり，参照が不要となった時点でFEP配下の磁気ディスクへ書き戻される。

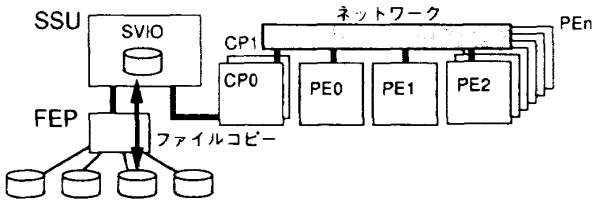

図7　SSU常駐化ボリューム機能

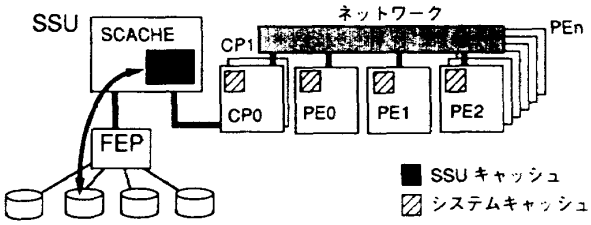

図8　SSU キャッシュ機能

(3)　プレステージング / ポストデステージング機能

ジョブ実行の高速化のために，ジョブ実行前に必要なデータを予め移動するプレステージング機能や，SSUの使用効率を向上するために，データが不要となった時点で速やかにデータをFEP上に書き戻すデステージング機能を設ける。

8.　運用管理機能

(1)　動的な PE 管理機能

数値風洞は，最大構成で200台程度のPEを接続

This document is provided by JAXA.

するため，消費電力量が大きい。このため，運用コストを考慮すると，システム負荷が低いときには不要なPEの電源を動的に切断したり，必要となった時点で電源を動的に投入できる機能が重要となる。

例えば，夜間の低負荷時には未使用のPEの電源切断を行う縮退運転機能がこれに該当する。

(2) 優先度制御機能

高優先度ジョブを即時に実行するため，実行中のジョブを中断するスワップ機能が必要である。スワップ機能は，(1)の縮退運転機能に対しても有効である。

(3) RAS機能

システムの信頼性・耐故障性の向上のために，RAS機能・障害解析機能が必要である。RAS機能としては，異常を検出したPEの切離しからシステムへの再組み込みまでの一連の処理を行えることが重要である。

(4) 障害解析機能

異常が発生したPEのダンプ取得，事象のトレース，エラー情報のロギング処理等の機能も必要である。

(5) 性能情報取得ツール

以下に示す性能情報を取得する機能が必要である。

(a) システム負荷 / 使用状況の取得 / 出力ツール

(b) チューナブルパラメータ化

(6) OS タイマの設定

各PE上のOSの設定時刻を極小の誤差とし，各PEの同時刻性を保証する仕組みも必要である。これは，並列処理プログラムの障害等を，Fortran等の事象トレースから解析するときなどに必要とされる。

9. おわりに

以上のように，数値風洞のOSは，多数のPEを効率的に管理し，並列処理の高速実行を支援する分散並列OSであると言える。分散並列OSは分散化と集中化を併存させており，経験を積みながら最適化 / 高速化を図る必要がある。

なお，本報告は，平成3年2月より開始された航技研・富士通の共同研究「数値風洞の開発研究」に基づいている。

参考文献

1) 三好　甫："CFDの推進に必要な計算機性能"，第8回航空機計算空気力学シンポジウム論文集，SP-13, pp.1-26, 1990年9月.

2) 高村守幸，岡田　信："CFD向け並列計算機のソフトウェア"，第8回航空機計算空気力学シンポジウム論文集，SP-13，pp.109-116, 1990年9月.

This document is provided by JAXA.

This document is provided by JAXA.