

NWTのユーザインタフェース実現方式

土屋雅子* 末松和代* 末松俊二*

島間晴夫** 森重博司** 山口靖** 軽部行洋**

A method of user interface for NWT

by

Masako TSUCHIYA, Kazuyo SUEMATSU, Shunji SUEMATSU

National Aerospace Laboratory

Haruo HATAMA, Hiroshi MORISHIGE, Yasushi YAMAGUCHI, Yukihiro KARUBE

Fujitsu Limited

ABSTRACT

Numerical Wind Tunnel(NWT) is a CFD-oriented parallel computer system with distributed memory, and the operating system of NWT is UNIX. In order to provide users much convenience and to maintain efficient functions of management on MSP it is preferable to use NWT with user interface of MSP which is the operating system up to this time.

In this paper we present how to construct user interface of MSP for NWT.

1. はじめに

数値風洞（以降、NWTと略記する）は要素計算機にベクトル計算機を配置する分散主記憶型並列計算機システムであり¹⁾、搭載OSはUNIXベースのOS（以降、NWT/OSと略記する）である²⁾。また、NWTは既設の大型電子計算機システム（FACOM VP2600：FEPと呼称する）をフロントエンドシステムとして結合した複合計算機システムを構成する。FEPのOSはNWTとFEPの緊密な連携をとるために、親和性の観点から、UNIXベースのOSを搭載する必要がある。しかし、航技研では10年来、富士通の独自OSであるMSP（以降、MSPと略記する）で運用しており、JCL等ユーザ側からみた使い勝手のノウハウを含めたMSPの資産

は膨大である。また、UNIXは不特定多数のユーザが混在して利用する大型汎用計算機システムのOSとしては、システムの管理運用機能が十分ではない。

以上の観点から、FEPには、MSPと富士通VPシリーズのUNIX-OSであるUXP（以降、UXPと略記する）の二つのOSを搭載し、ユーザインタフェースはMSPビューとする運用を検討した。MSPと、NWT/OSおよびUXPのUNIX-OS（以降、NWT/OSとUXPの両者を新運用OSと呼称する）では、ユーザビューにおいて、種々な相違点がある。MSPビューとするためには、両OSの相違を吸収するための各種インタフェースが必要となる。

本報告は、これらのインタフェースの実現方式の検討結果について述べる。

2. NWTのハードウェア構成概念図

図1に、NWTのハードウェア構成概念図を示

*航空宇宙技術研究所

**富士通株式会社

す。NWTは、制御処理装置、並列に配した演算処理装置、およびそれらを相互接続するクロスバネットワークより構成される。制御処理装置はシステム記憶を介して大型電子計算機システムのFEPと接続される。ジョブ入出力のフロントエンド処理やNWTで実行するジョブ（以降、NWTジョブと呼称する）の翻訳・結合編集、TSS処理等はFEPが分担する。NWTは、そのハードウェアが有する超高速処理性能を十分に引き出すために、大規模並列ベクトル計算処理の実行エンジンとして専念する運用とする。

3. ユーザビューにおけるMSPとUNIXの主な相違点と必要なインタフェース機能

(1) ジョブの記述形式

航技研のシステムでは、MSPが有するカタログ・プロシジャ（以降、カタプロと呼称する）機能を駆使した航技研独自のカタプロを多数用意している。ユーザはこのカタプロを利用すれば、目的とするジョブの記述が非常に容易である。UNIXでは、MSPのジョブに対応する処理単位概念はなく、バッチリクエストというプロセスの単位が、MSPのジョブを構成するジョブステップに対応している。カタプロ記述形式で投入されたNWTジョブが、ジョブステップの実行順序を違えることなく、順番に新運用OS上で、バッチリクエストとして処理されるためには、バッチリクエストを記述するシェルスクリプトをカタプロから生成するインタフェース機能が必要となる。

(2) ファイル・システム

MSPのファイルは区分編成、順編成等、種々のファイル編成や任意のブロック長をもつが、UNIXのファイルは単なるバイト列であり、固定長ブロックのデータの集まりである。また、FORTRAN記録の各種レコード形式におけるフォーマットも、MSPとUNIXでは相違する。このため、MSPのファイルとUNIXのファイルを双方向に変換するためのインタフェース機能が必要となる。

(3) 出力リストの取り出し方式

MSPでは、ジョブの出力リストもジョブとし

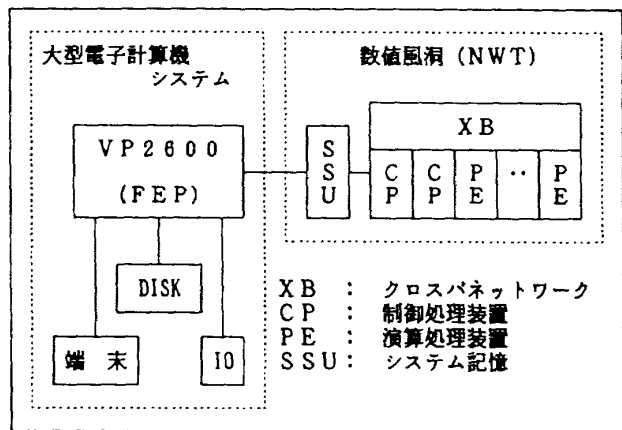


図1 NWTのハードウェア構成概念図

て管理され、ジョブ単位に1つの出力待ちジョブとしてスプール・ファイルに出力される。UNIXの場合には、出力リストはプロセス単位に通常の入出力ファイルと同様に、ファイルとして出力される。複数ステップからなるFORTRANプログラムのジョブをUNIXで実行すると、出力リストとして、翻訳結果リスト、結合編集結果リストおよび実行処理結果リスト等、複数のファイルが作成される。これをMSPビューとするには、これらのファイルをジョブとして統合して扱うためのインタフェース機能が必要となる。

(4) 文字データ形式

ソースプログラムやカードイメージデータ等の文字データの形式は、MSPではEBCDICコード、UNIXではASCIIコードである。従って、これらのデータが両OSを遷移するためには、コード変換のためのインタフェース機能が必要となる。

4. NWTジョブ処理の概要

図2は、NWTジョブ処理の流れの概念図である。NWTジョブは通常、翻訳処理（複数ステップ構成も可）、結合編集処理、実行処理（複数ステップ構成も可）の構成でシステムに投入される。同図では、MSPビューのユーザインタフェースでシステムに投入されたNWTジョブが、MSP、UXPおよびNWT/OSの各種OS処理フェーズによるインタフェースを取りながら、実行していく様子を示している。同図に基づいて、以下に、NWTジョブの処理の概要を示す。なお、NWTジョブの実行に必要なファイルの転送方式につい

ては、第5章に示す。

4.1 NWTジョブ投入処理

MSPが管理する各種のTSS端末から、ユーザがNWTジョブを投入する処理フェーズである、本処理フェーズがMSPビューのユーザ・インタフェーズで行えるためには、また、TSSの応答性の観点からも、NWTジョブの実行に必要なファイルはMSPで管理する運用とする。また、NWTと直接会話しないで、NWTへの処理依頼をMSPから伝達するためには、以下の機能を有するTSSコマンドが必要となる。

(1) NWT SUBコマンド

本コマンドは、MSPからNWTジョブの実行起動を指示するコマンドである。

(2) NWT CANコマンド

本コマンドは、NWT上で実行待ち、実行中状態のNWTジョブをキャンセルするコマンドである。

(3) NWT OUTコマンド

NWTジョブの出力リストはバッチリクエストごと、またOS処理フェーズごとに、それぞれ独立のファイルに出力され、複数の出力結果が生じる。本コマンドは、これらの複数ファイルを統合し、ジョブ検索やプリンタからの取り出しを指示するコマンドである。なお、第3章(3)に示したMSPとUNIXの相違は、本コマンドの実行により吸収する。

(4) N F O R Tコマンド

本コマンドは、UXPで実行する翻訳、結合編集処理の起動と処理結果の応答を、MSPからコマンドモードで指示するコマンドである。

(5) N W T S Tコマンド

本コマンドは、ユーザごとのNWTジョブ実行状況をセッション端末に表示することを指示するコマンドである。

(6) S T A T Eコマンド

本コマンドは、NWTジョブを実行待ち、実行中、実行結果出力待ち状況に分け、NWTシステムの混雑状況をMSPの端末に表示することを指示するコマンドである。

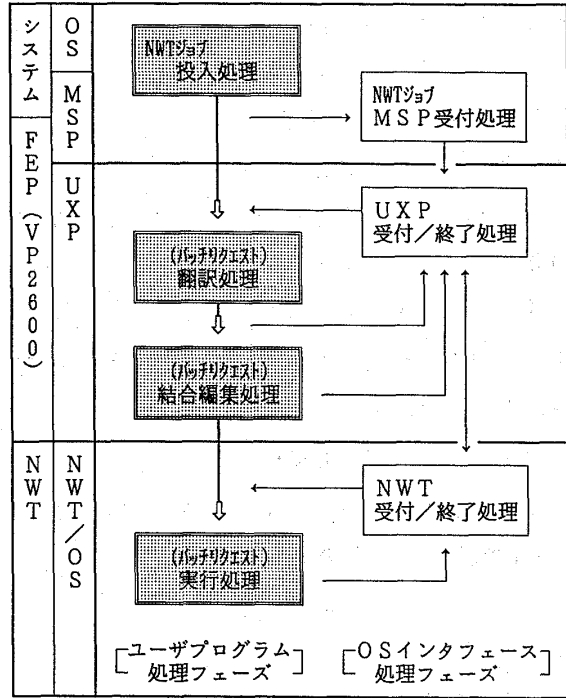


図2 NWTジョブ処理の流れの概念図

4.2 MSP受付処理

本処理フェーズは、MSPのOS処理フェーズである。ここでは、NWT SUBコマンドを受けて、NWTジョブの受付処理を行う。本処理フェーズの主な処理概要を以下に示す。

(1) カタプロ確認/検査処理

カタプロ必須パラメータと構文、および全入出力ファイルの割当の確認/検査を行う。ジョブ記述上のエラーによるジョブ再投入を少なくするため、エラー処理は後述のUXP受付処理で集約し、エラー情報を詳細に出力し得るよう、本処理フェーズでは、MSPでなければ確認できない処理のみを行う。エラー発生の場合には、その旨のメッセージをユーザごとのメッセージ・ファイルに出力する。

(2) カタプロ変換処理

カタプロ記述形式をUNIXの中間シェルスクリプトの形式に変換する。図3に、カタプロの中間シェルスクリプト変換例を示す。なお、第3章(1)に示したMSPとUNIXの相違は本処理で吸収する。

(3) ジョブの転送

MSPで正常に受け付けたNWTジョブをUXPに自動転送する。

(4) ファイル転送のセットアップ

カタプロ指定ファイルをU X Pに送信するためのセットアップを行う。

4.3 U X P 受付/終了処理

本処理フェーズはU X PのO S処理フェーズである。M S Pから転送されたN W Tジョブを確認/検査し、バッチリクエストを生成するための受付処理およびU X Pで実行したバッチリクエストの終了処理を行う。本処理フェーズの主な処理概要を以下に示す。

(1) シェルスクリプトの展開

U N I X中間シェルスクリプトをシェルスクリプトに自動展開し、同時に構文の確認/検査を行う。エラー発生の場合には、その旨のメッセージをユーザごとのメッセージ・ファイルに出力する。また、U X PおよびN W T/O Sで標準入出力するデータファイルやライブラリの定義ならびにシステムパラメータの設定を行う。

(2) バッチリクエストのスケジューリング処理

N W Tで実行するバッチリクエストを除く全てのバッチリクエストはF E Pで実行されるようにスケジューリングする。

(3) ファイル転送のセットアップ

バッチリクエストが入出力するファイルをM S Pと送/受信するためのファイル転送をセットアップする。

(4) バッチリクエストの転送

N W Tで実行するバッチリクエストをN W Tに転送する。

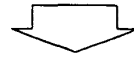
4.4 言語処理

本処理フェーズは、N W T用の並列フォートラン言語で記述されたユーザプログラムの翻訳処理をU X P上で行う処理フェーズである。ソースプログラムファイルを入力し、実行結果としてオブジェクトモジュール・ファイルを出力する。

4.5 結合・編集処理

本処理フェーズは、言語処理フェーズに続く、結合編集をU X P上で行うユーザプログラムの処理フェーズである。先行の言語処理フェーズが出

```
//X01PASS NJOB XXXX
// EXEC NFORTC, PARAM='-Os,-i,-Pt',
// SF='X01.A.FORT77', SINC='X01.A.INCLUDE'
// EXEC NLIED, OEP='X01.A.LOAD(MAIN)'
// EXEC NGO, EF='X01.A.LOAD', EMB=MAIN, DF='X01.A.DATA'
// EXPAND NUSDKR, RNO=01, FILE='X01.FT01.INPUT', CONV=YES
// EXPAND NUSDKW, RNO=02, FILE='X01.FT02.OUTPUT', ORG=PO,
// DISP=NEW, SPACE=(2,2)'
```



```
njob -r X01S767 -q XXXX -m X01S767
      _NWT.NWJ.X01S767.SH_nb
nfortc -p'-Os,-i,-Pt' -i $TMP/nwj/X01S767/include
      X01S767_X01.A.FORT77(A1) _nn.f
      X01S767_X01.A.FORT77(B1) _nn.f
nlied -o X01S767_X01.A.LOAD(MAIN) _nb
ngo -f X01S767_X01.A.LOAD(MAIN) _nb
     < X01S767_X01.A.DATA _nn
nusdkr -n 01 X01S767_X01.FT01.INPUT_nb
nusdkw -n 02 X01S767_X01.FT02.OUTPUT_nb
nuswk -n 08
ngoend
```

図3 カタプロの中間シェルスクリプト変換例

力したオブジェクトモジュール・ファイル、ユーザ・ライブラリ・ファイルならびにシステム・ライブラリ・ファイル等を入力し、実行結果としてロードモジュール・ファイルを出力する。

4.6 N W T 受付/終了処理

本処理フェーズは、N W Tで実行するバッチリクエストの受付/終了処理を、N W T/O S上で実行するO S処理フェーズである。

(1) バッチリクエストのスケジューリング処理

実行待ちバッチリクエストをどの様にN W Tで実行させるか、また、使用P Eをいかに割当るか等のスケジューリング処理を行う。

(2) ファイルのステージング処理

N W T実行時入出力ファイルをS S UとU X Pファイル間において複写/退避(ステージング/デステージングという)のスケジューリングを行う。

4.7 N W T 実行処理

本処理フェーズは、ユーザプログラムの処理フェーズであり、並列ベクトル計算の実行処理をN W Tで実行する。S S U上の、ロードモジュール・ファイルと入出力ファイルをアクセスして実行する。なお、入出力するバイナリーデータの内部形式は、N W TではI E E E形式、F E PではI B M形式である。この相違はF O R T R A Nライ

ブラリが吸収する。

5. ファイル転送実現方式の概要

翻訳、結合編集処理等のFEP実行時に入出力するファイルの転送方式と、NWT実行時入出力ファイルの転送方式の2種類のファイル転送方式を実現する。また、第3章(2)および(4)に示したファイルシステムおよび文字形式の相違については、これらの転送処理の中で吸収する。

5.1 FEP実行時入出力ファイルの転送処理

図4に、FEP実行時入出力ファイルの転送方式を示す。UXPの下でFEP実行時に入出力するファイルは、MSPから転送し、UXP管理の磁気ディスクボリュームにUXP中間ファイルとして格納する。FEP実行時には、このUXP中間ファイルをアクセスし、実行データを入出力する。また、FEP実行終了後、この入出力データをMSPに格納する必要がある場合には、UXP中間ファイルをMSPに転送する。以下にファイル転送の実現方式の概要を示す。

(1) ファイル送信処理

①NWTSUBコマンドによりユーザジョブを受け付けると、システムは送信データ作成ジョブを起動する。

②送信データ作成ジョブは、入力ファイルを読み込み、M-M間経路(MSPメモリ-UXPメモリ間転送経路)を利用して、UXP側のファイル受信デモンに送信する。

③ファイル受信デモンは、受信したデータをUXPが管理するUXP中間ファイル(入力データ用)に格納する。

(2) ファイル受信処理

④ジョブ終了時にファイル後処理バッチリクエストを起動する。バッチリクエストでは、ファイル受信ジョブをM-M間経路で、MSP側の通信ジョブリーダに転送する。

⑤通信ジョブリーダは、ファイル受信ジョブを起動する。

⑥後処理バッチリクエストは、UXP中間ファイル(出力データ用)を読み込み、ファイル受信ジ

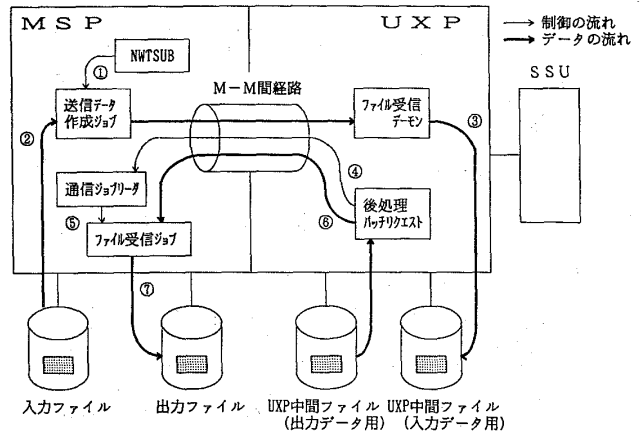


図4 FEP実行時入出力ファイルの転送方式

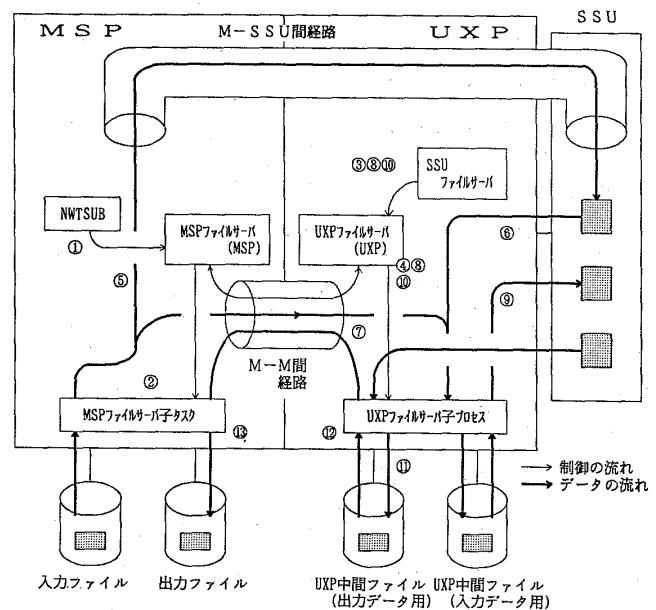


図5 NWT実行時入出力ファイルの転送方式

ョブに転送する。

⑦ファイル受信ジョブは、受信したデータをMSPの出力ファイルに格納する。

5.2 NWT実行時入出力ファイルの転送処理

図5に、NWT実行時入出力ファイルの転送方式を示す。NWTにおける大規模並列ベクトル計算実行時には、膨大なデータ量をアクセスする。このアクセスに要する入出力経過時間を最小にするため、磁気ディスク装置よりデータ転送性能が格段に優れたSSUが補助記憶装置として利用される。ファイルをSSUに格納するステージング処理、また、実行結果としての出力ファイルをMSPに送信するために、SSUから退避するデステージング処理に必要なファイル転送の実現方式の概要は以下のとおりである。

(1) ステージング処理

- ① NWT SUBコマンドによりユーザジョブを受け付けると、MSPにおけるファイル転送処理を制御するMSPファイルサーバに通知する。
- ② MSPファイルサーバは、ユーザジョブ毎に通信経路を指定し、MSPファイルサーバ子タスクを生成する。
- ③ SSUにおけるステージング処理を制御するSSUファイルサーバよりステージング要求があると、UXPにおけるファイルの転送処理を制御するUXPファイルサーバに通知される。
- ④ UXPファイルサーバは、通信経路を指定してUXPファイルサーバ子プロセスを起動する。UXPファイルサーバ子プロセスはMSPファイルサーバ子タスクにデータ転送を要求する。
- ⑤ MSPファイルサーバ子タスクは、入力ファイルを読み込み、M-SSU間経路(MSPメモリー-SSU間転送経路)を利用して、UXP配下のSSUに転送する。
- ⑥ UXPファイルサーバ子プロセスは、⑤の処理により転送されたSSU上のデータをUXP中間ファイル(入力データ用)に複写する。
- ⑦ SSUが使用できない状態になったら、MSPファイルサーバ子タスクは、経路をM-M間経路に変更し、UXPファイルサーバ子プロセスに直接データ転送する。このデータを、UXPファイルサーバ子プロセスは、⑥のデータに続けて格納する。
- ⑧ SSUファイルサーバより入力要求があると、UXPファイルサーバは、UXPファイルサーバ子プロセスにデータ転送処理を依頼する。
- ⑨ UXPファイルサーバ子プロセスは、UXP中間ファイル(入力データ用)よりデータを読み込み、SSUに転送する。
- (2) デスステージング処理
- ⑩ SSUファイルサーバより出力要求があると、UXPファイルサーバは、UXPファイルサーバ子プロセスにデータ転送処理を依頼する。
- ⑪ UXPファイルサーバ子プロセスは、SSUよりデータを読み込み、UXP中間ファイル(出力データ用)に転送する。

⑫ UXPファイルサーバ子プロセスは、UXP中間ファイル(出力データ用)よりデータを読み込み、M-M間経路を利用して、MSPファイルサーバ子タスクにデータを転送する。

⑬ MSPファイルサーバ子タスクは受信したデータを格納ファイルに出力する。

7. おわりに

本稿では、UNIXベースのOSを搭載しているNWTを、MSPビューのユーザインタフェースで利用するための実現方式を示した。これにより、ユーザは、従来と同等の利用法で新システムのNWTを利用できる見通しとなった。また、管理運用上でも、UNIXには、いまだ十分でない機能をMSPで代替することが可能となった。

汎用大型計算機システムのOSとして、将来的には、豊富なネットワーク・通信機能を持ち、オープン・インタフェースとして普及しているUNIXが今後、主流となるとも考えられる。これに対しては、UNIXの動向をよく把握して、NWTを直接利用するためのUNIXのユーザインタフェースについて、検討する予定である。

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, 吉岡, 他: "数値風洞のハードウェア" 第9回航空機計算空気力学シンポジウム論文集, SP-16, 1991年12月
- 2) 福田, 末松, 他: "数値風洞のオペレーティングシステム", 第9回航空機計算空気力学シンポジウム論文集, SP-16, 1991年12月