

JAERI-Tech  
97-026



UNIX-WSを利用した新JT-60実験  
データベースシステムの開発

1997年6月

戸塚俊之・栗原研一

日本原子力研究所  
Japan Atomic Energy Research Institute

本レポートは、日本原子力研究所が不定期に公開している研究報告書です。  
入手の間合わせは、日本原子力研究所研究情報部研究情報課（〒319-11 茨城県那珂郡東海村）あて、お申し越してください。なお、このほかに財団法人原子力公済会資料センター（〒319-11 茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

This report is issued irregularly.  
Inquiries about availability of the reports should be addressed to Research Information Division, Department of Intellectual Resources, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

© Japan Atomic Energy Research Institute, 1997

編集兼発行 日本原子力研究所  
印 刷 (株)高野高速印刷

## UNIX-WSを利用した新JT-60実験データベースシステムの開発

日本原子力研究所那珂研究所核融合装置試験部

戸塚 俊之・栗原 研一

(1997年5月27日受理)

近年の計算機技術の急速な進展を反映し、那珂研究所に設置の原研情報システムセンターの大型汎用計算機が撤去されワークステーションを用いて分散処理を行うシステムへとその構成が大きく変更されることとなった。そのシステムで核となるのは、解析計算やデータ管理を統括する複数のワークステーションである。この変更に伴い、大型汎用計算機に構成されていたJT-60実験データベースシステムについても新システムに対応して新たに開発する必要が生じた。

新しいJT-60実験データベースシステムでは、

- ①ネットワークを介しデータの送受信を行えること。
- ②今まで大型汎用計算機に構築されてきた各種解析コード等の資源をそのまま利用できること。

という要求が与えられていた。また、いくつかの設計方針を採用した。即ち、

- ①新しいJT-60実験データベースシステムではUNIXワークステーションの標準搭載機能を極力利用して作成することにした。これは、異機種への移行を容易にするためである。
- ②UNIX特有のディレクトリ構造とデータベース構造が同じ階層構造であるという類似性を利用し、特別なデータ検索や排他のためのソフトウェアを作成せずに標準的なUNIX-OSの機能を用いることにした。これにより、開発工数の低減が図れる。
- ③Xウィンドウを積極的に利用した効率的なデータベース運用管理機能とする。

本報告書では、このようにして構築したデータベースシステムについて機能詳細設計内容、性能評価結果他を記述する。

Development of a New Database System  
for JT-60 Experiments Utilizing UNIX Workstations

Toshiyuki TOTSUKA and Kenichi KURIHARA

Department of Fusion Facility  
Naka Fusion Research Establishment  
Japan Atomic Energy Research Institute  
Naka-machi, Naka-gun, Ibaraki-ken

(Received May 27, 1997)

JAERI computer system has been newly composed of distributed UNIX WS computers in place of a centralized general Purpose computer, having been considered as a basic invariant structure for data processing with high quality for long.

Characteristics of the new system are as follows,

- (1) Subroutine names and sequence of the variables are keep as the identical formalism with those in the former system.
- (2) Experimental data are supposed to be utilized for the JT-60 analysis through a network of Ethernet.
- (3) As a hierarchical data structure in the database system, "directory" structure in the UNIX-OS is utilized just as it is.
- (4) Database management is fully improved using the "X-window" tools.

In the report, Dvelopment of the new database system for JT-60 utilizing UNIX Workstations is presented. The tests to generate a database and to read from the database system aer conducted and its results are disussed for evaluation of the new system.

Keywords : JT-60, Database, UNIX, Workstation, Network, Computer System, Ethernet

## 目 次

|     |                                    |    |
|-----|------------------------------------|----|
| 1.  | はじめに .....                         | 1  |
| 2.  | 新実験DB管理システムの基本構想 .....             | 2  |
| 2.1 | 新JT-60実験DBのシステム構成検討 .....          | 3  |
| 2.2 | 新JT-60実験DB管理システムのハードウェア構成と性能 ..... | 4  |
| 2.3 | 全系DB管理システムのソフトウェア構成と機能 .....       | 6  |
| 3.  | 全系DBの構成と管理の基本設計 .....              | 8  |
| 3.1 | 全系DB管理の構造検討 .....                  | 8  |
| 3.2 | UNIXディレクトリ構成を利用したDB構成 .....        | 11 |
| 4.  | 全系DB創成機能 .....                     | 15 |
| 4.1 | 全系DBサーバからDBサーバへのデータ転送方法の検討 .....   | 15 |
| 4.2 | 全系DB創成管理機能 .....                   | 16 |
| 4.3 | 全系DB創成機能 .....                     | 16 |
| 4.4 | 過去のデータベース再格納機能 .....               | 17 |
| 5.  | ネットワークを介したデータベースへのアクセス .....       | 19 |
| 5.1 | 全系DBハンドリング・パッケージ.....              | 20 |
| 5.2 | ネットワーク・アクセス・サーバの機能 .....           | 22 |
| 5.3 | ローカル・アクセス・ルーチンの機能 .....            | 22 |
| 6.  | 全系DB運用管理機能 .....                   | 23 |
| 7.  | 性能評価と考察 .....                      | 25 |
| 7.1 | 全系DB創成処理の実行時間 .....                | 25 |
| 7.2 | ディレクトリ階層の深さの違いによる全系DBアクセス性能 .....  | 26 |
| 7.3 | ネットワークを介した全系DBのアクセス速度 .....        | 26 |
| 8.  | まとめ .....                          | 27 |
|     | 謝 辞 .....                          | 28 |
|     | 参考文献 .....                         | 28 |
| 付録1 | 新全系DBファイルフォーマット .....              | 29 |
| 付録2 | DB管理テーブルフォーマット .....               | 33 |
| 付録3 | 旧全系DBファイルから新全系DBファイルへの変換例 .....    | 34 |
| 付録4 | IBMフォーマットとIEEEフォーマットのデータ表現形式 ..... | 39 |
| 付録5 | 全系DBハンドリング・パッケージの問題点 .....         | 40 |
| 付録6 | 基本サブルーチン通信フォーマット .....             | 42 |
| 付録7 | 資格検査テーブルの構造 .....                  | 46 |
| 付録8 | 全系DB運用管理画面.....                    | 47 |

## Contents

|   |    |
|---|----|
| 1. Introduction .....   | 1  |
| 2. Basic Conceptual Consideration on the<br>New JT-60 Database Management System (DBMS) ..... | 2  |
| 2.1 Investigation on the Database System Structure .....                                      | 3  |
| 2.2 Configuration on the Hardware .....   | 4  |
| 2.3 Configuration on the Software .....   | 6  |
| 3. Design of the Database System .....  | 8  |
| 3.1 Investigation on the Data File Structure .....  | 8  |
| 3.2 Hierarchical File Structure Utilizing UNIX Directories .....                              | 11 |
| 4. Database Generation .....  | 15 |
| 4.1 Investigation on the Data Transfer .....  | 15 |
| 4.2 Management System of the Database Generation .....  | 16 |
| 4.3 Procedures of the Database Generation .....   | 16 |
| 4.4 Restore of the Old Database File .....  | 17 |
| 5. Interface to the Database through a Network .....  | 19 |
| 5.1 Data Handling Interface Programs .....  | 20 |
| 5.2 Functions of "Network Access Server" .....  | 22 |
| 5.3 Functions of "Local Access Routines" .....  | 22 |
| 6. Maintenance and Operation of the Database System .....                                     | 23 |
| 7. Evaluation of the Performances .....   | 25 |
| 7.1 Execution Time of the Database Generation .....   | 25 |
| 7.2 Time Comparison to Access to File with the different Directory Depths .....               | 26 |
| 7.3 Time Comparison to Read the Database through a Network .....                              | 26 |
| 8. Concluding Remarks .....   | 27 |
| Acknowledgments .....   | 28 |
| Reference .....   | 28 |
| Appendix 1 Database File Formats .....  | 29 |
| Appendix 2 DB Control Table Formats .....   | 33 |
| Appendix 3 An Example of the Old and New DB Formats .....                                     | 34 |
| Appendix 4 IBM and IEEE Formats .....   | 39 |
| Appendix 5 Problem and Solutions in the Data Handling Interface Programs .....                | 40 |
| Appendix 6 Message Formats .....  | 42 |
| Appendix 7 The Structure of Authorization Check Table .....                                   | 46 |
| Appendix 8 A Screen Format for Operating and Maintenance .....                                | 47 |

## 1. はじめに

JT-60の実験・運転により発生する大量の放電に関する放電制御データは、制御システムの中核である全系制御設備（以下「全系」と称す。）で一旦収集される。その後直ちに全系計算機に収集された放電制御データは、計測システムで収集された計測データと共に大型汎用計算機（FACOM M780:以下「FEP」と称す。）に転送され、浮動小数点化等の変換処理が施されJT-60実験データベース（以下「実験DB」と称す。）として登録される。（このうち、全系計算機によって収集された放電制御データをデータベース化したものを以下「全系DB」と称す。）以降、データ解析は、この実験DBを基に進められるという利用形態を採ってきた。

近年の小型計算機技術の急速な進展を反映し、これまで那珂研究所に設置されていた、大型汎用計算機が撤去され、ワークステーションを用いて分散処理を行うシステムに置き換えることとなった。そこで、データ管理を統括するワークステーション上に新たな実験DBシステムを構築することが要求された。具体的な要求は、以下の通りである。

- ① FEPに構築されてきた実験DBをオペレーティングシステム(OS)等の全く環境の異なるUNIXワークステーションに移行するためには、新たな実験DBシステムを構築する必要がある。
- ② ネットワークを利用した分散処理計算機システム化に伴い計算機間通信によるデータベースの取得が不可欠となり、これに対応するソフトウェアを作成する必要がある。
- ③ これまでFEP上に蓄積されてきた実験DBを新システム上に反映させるための変換機能を作成する必要がある。

これらの要求を満たすことに加え、以下に示す基本方針の下に新しいJT-60実験データベースシステムを開発することとした。

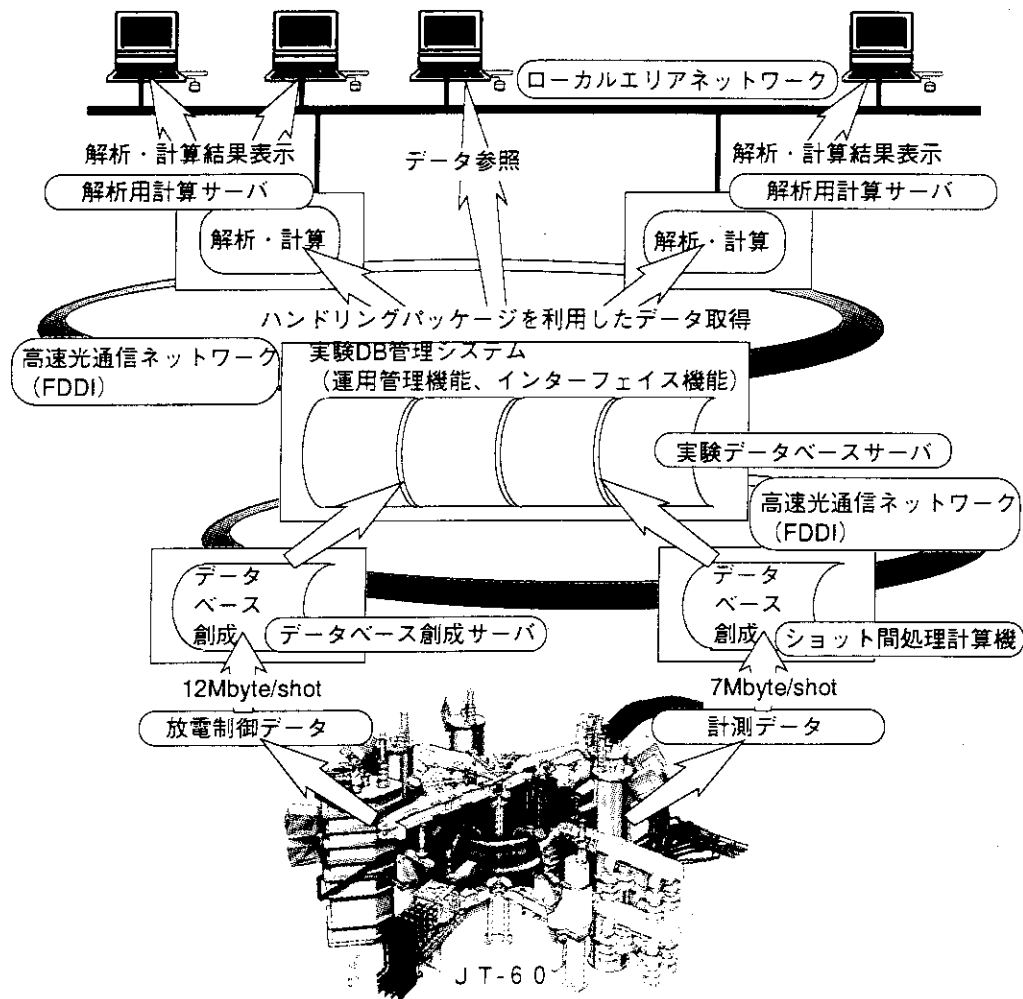
- ① UNIXワークステーションの標準搭載機能を極力利用して本システムを作成することにした。これは、異機種への移行を容易にするためである。
- ② UNIX特有のディレクトリ構造とデータベース構造とが同じ階層構造であるという類似性を利用し、特別なデータ検索や排他のためのソフトウェアを作成せずにUNIX-OSに任せることにした。これにより、開発工数の低減が図れる。
- ③ X-Windowを積極的に利用して効率的なデータベース運用管理機能とする。

本報告では、第2章で、UNIX計算機上に構築する新しいJT-60実験データベースシステムの基本構想を、第3章では、全系DBの基本構成と実機導入に関する全系DBの設計変更経緯を、第4,5及び6章では、新JT-60実験データベースシステムに関するソフトウェア機能を、第7章で性能評価結果を示し、それに対して考察を加える。最後にまとめを行う。

## 2. 新実験DB管理システムの基本構想

第2.1図に示すように、新JT-60実験DB管理システムは、分散化処理体形を基本に高速光通信ネットワークFDDI（Fiber Distributed Data Interface）で接続された複数の計算機システムで構成される。ここでは、JT-60実験DBを格納・管理するための大容量の磁気ディスクを持った実験データベースサーバを中核にJT-60の実験に伴い発生する放電制御及び計測のデータ収集用のネットワークと実験DBを利用する解析用計算サーバの接続されるデータサービス用のネットワークとの2つがそれぞれ専用のFDDIで接続される。

解析用計算サーバは、ネットワークを介して実験データベースサーバに蓄積された実験DBのデータを取得する。この時、実験DB取得専用のハンドリングパッケージを使用する。解析・計算結果は、ローカルエリアネットワークに接続されたワークステーションやX端末で表示できる。このように新JT-60実験DB管理システムは、データ収集、蓄積・管理、解析・計算及び表示をそれぞれ別個の専用の計算機で分散処理する構成を採ることを基本として開発することとした。



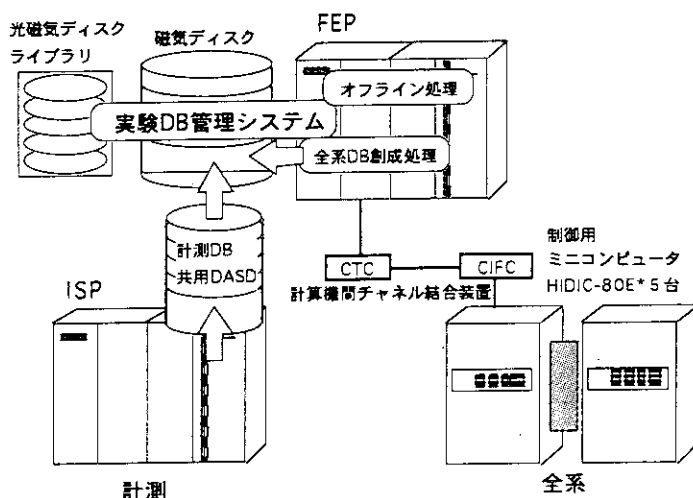
第2.1図 新JT-60実験DB管理システム概念図



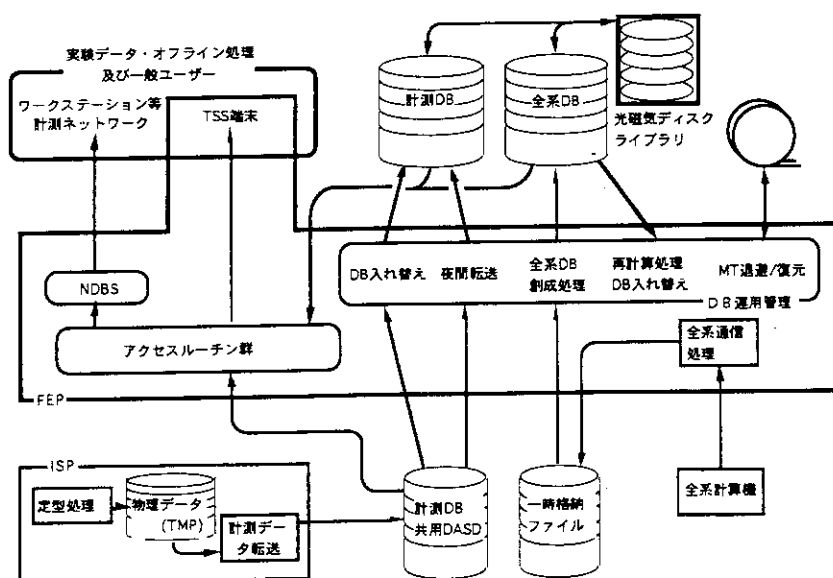
## 2.1 新JT-60実験DBのシステム構成検討

第2.2図示すように、放電制御系のデータは、JT-60各設備から全系制御設備の制御用ミニコンピュータ（HIDIC-80E）に収集され専用の計算機間チャンネル結合装置を介してFEPに転送されていた。転送されたデータは、FEP上の全系DB創成処理によりデータベース化される。また、計測データは各計測装置からデータ処理設備のショット間処理計算機（ISP）に収集され、データベース化を行い共有ディスク装置によりFEPに転送されていた。

FEP上には、実験DB管理システムが備えられていて、それぞれのデータベースを管理する。また、データ利用のためのユーティリティであるデータ・アクセス・ルーチンや各種システム管理機能が備えられていた（第2.3図参照）。これらを利用し、JT-60実験データのオフライン処理や他の解析処理が行われていた。従って、FEP撤去に伴い、このFEP上の実験DB管理システムの再構築や同等のJT-60実験データの解析用計算のための計算機環境の整備を図る必要が生じた。



第2.2図 旧JT-60実験DB管理システムの構成



第2.3図 旧実験DBの利用形態

まず、システムに備えるべき要件及びユーザ側からの要請とそれらに対応した開発基本方針は次の通りである。

- (1) FEPで実験データベースを参照・解析していたユーザは、FEP撤去後は、その代替計算機として解析用計算サーバ、または独自のワークステーションの何れかを選ぶこととなる。従って、これらの計算機から実験DBを利用できることが必要である。  
これについては、ネットワークに接続されたワークステーション等からネットワークを介した実験DBを取得するための専用ハンドリング・ルーチンを開発し提供することで対応することとした。
- (2) 新たなシステムへの移行は、JT-60の実験遂行に支障の無いよう円滑に行うことが必要である。  
これについては、UNIX-OSの基本機能を極力利用して独自に製作するプログラムを極力少なくしソフトウェア開発に要する工数を少なくすることとした。また、実機導入以前にプログラムの完成度を極力高めるため、試験システムを用いた開発を行うこととした。さらに、実験DBのユーザに対しては、専用ハンドリング・ルーチンを早期に提供し、実験DB利用ソフトでの確認を前もって出来るように考慮した。
- (3) また、JT-60の実験遂行に支障の無いようシステムの拡張性や信頼性を充分確保するため、比較的安価で拡張性に優れたUNIXワークステーションと信頼性の向上が図れるRAID<sup>\*1</sup>方式のディスクアレイシステムを採用することにより、コストパフォーマンスの高いシステムを構築するものとした。
- (4) 利用頻度の高いショットのデータベースの利用が効率良く行えるよう、これらのデータベースは、15,000ショット（7年間分に相当）分のデータベースが磁気ディスクと光磁気ディスクライブラリ装置に格納出来ることとした。

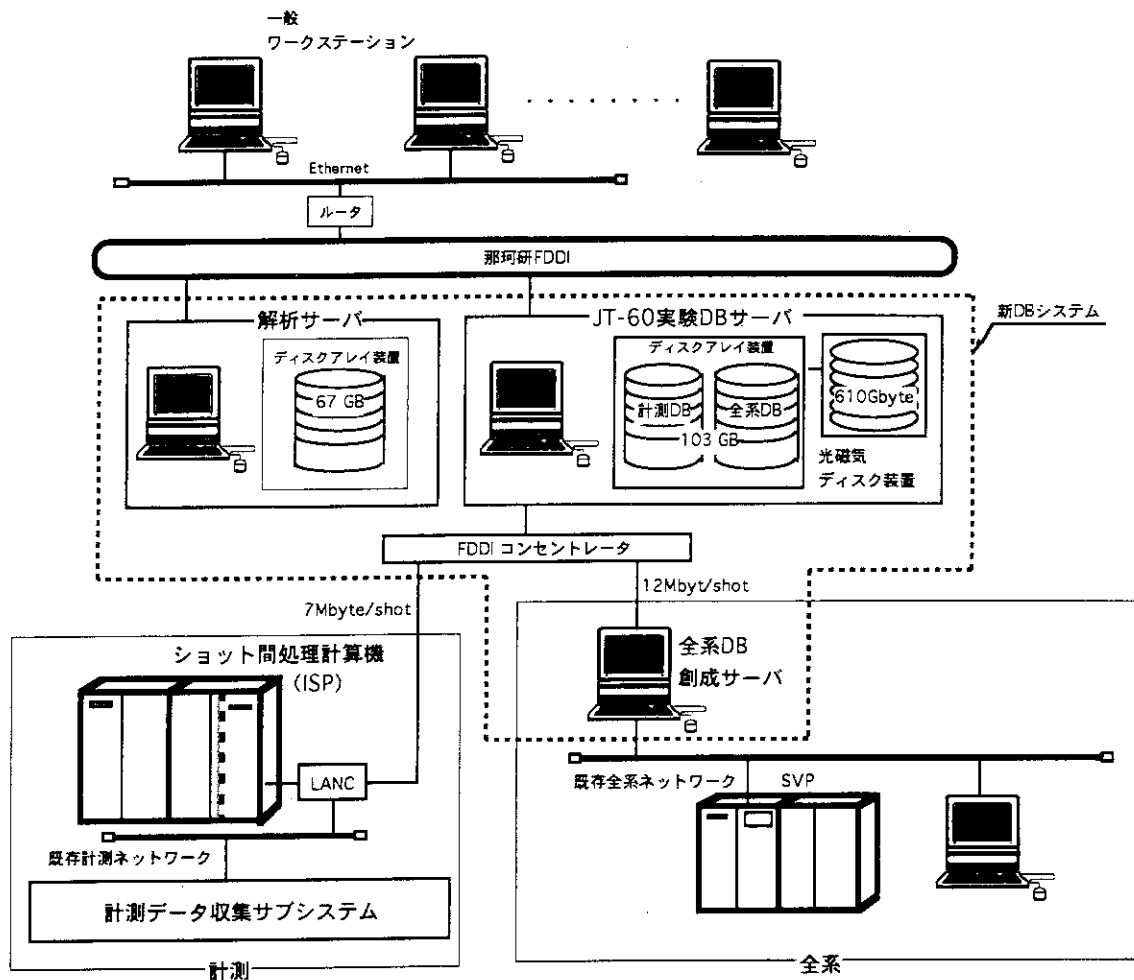
## 2.2 新JT-60実験DB管理システムのハードウェア構成と性能

第2.1節の検討結果から新しいJT-60実験DB管理システムのハードウェアとして、第2.4図に示す構成を採った。

新しい実験DB管理システムは、実験DBを格納・管理するJT-60実験データベースサーバ（以下「DBサーバ」と称す。）と実験DBを利用して各種解析処理をおこなう解析サーバ、全系のデータベース創成を行う全系DB創成サーバ及び計測データの収集及びデータベース創成を行うショット間処理計算機（ISP）が専用のFDDIで接続され構成されている。

DBサーバ、解析サーバ、全系DB創成サーバのハードウェアの仕様は第2.2表に示す。

\*1 RAID : Redundant Array of Inexpensive(or Independent) Disksの略で、当初、小型で安価なディスク・ドライブを複数組み合わせることで単一の大型で高価なディスクを越える性能を引き出すために開発されたディスク記憶方式を言うが、現在では、ミラーリングや専用パリティの付加などにより、データの誤り検出と訂正など、信頼性向上を図る技術として普及しつつある。



第2.4図 新しいJT-60実験DB管理システムのハードウェア構成

第2.2表 ハードウェアの仕様

|                | JT-60実験DB サーバ              | 解析サーバ                                   | 全系DB創成サーバ                       |
|----------------|----------------------------|---|---------------------------------|
| 機器名称           | fujitsu S-4/1000E          | IBM POWER 並列サーバ<br>9076-SP2             | fujitsu S7/300Uモデル170           |
| CPUタイプ         | 85MHz SuperSPARC2<br>2CPU  | 77MHz POWER2 <sup>7</sup> プロセッサ<br>4CPU | 167MHz UltraSPARC               |
| 主記憶容量          | 256MB                      | 1,024MB                                 | 256MB                           |
| 内蔵ディスク         | 37.8GB                     | 18.0GB                                  | 2.1GB                           |
| 外部ディスク         | ディスクアレイ装置モデル112<br>113.4GB | RAIDIANアレイ・ディスク<br>67.5GB               | マルチディスクパック<br>2.1GB*12 (25.2GB) |
| 光ディスクライブラリ     | 87.1GB*7 (609.7GB)         |   |                                 |
| ネットワークインターフェイス | FDDI*2, Ethernet*4         | FDDI*2, Ethernet                        | FDDI, Ethernet                  |

### 2.3 全系DB管理システムのソフトウェア構成と機能

分散処理体系で構成された新しいJT-60実験DB管理システムに対応するための全系DB管理システムに必要なソフトウェアの機能を第2.3表に構成を第2.5図に示す。

全系DB創成サーバは、SVPに収集された放電制御データを一時格納ファイルという生データの集合ファイルで受信する。全系DB創成サーバは、受信した一時格納ファイルをそのままの形式で保管すると共に全系DB創成機能によりデータベース化を行いJT-60実験DBサーバに全系DBを格納する。また、旧DBフォーマットで記録された過去の全系DBをMTから再現し新全系DBに変換処理を施しJT-60実験DBサーバに格納する。

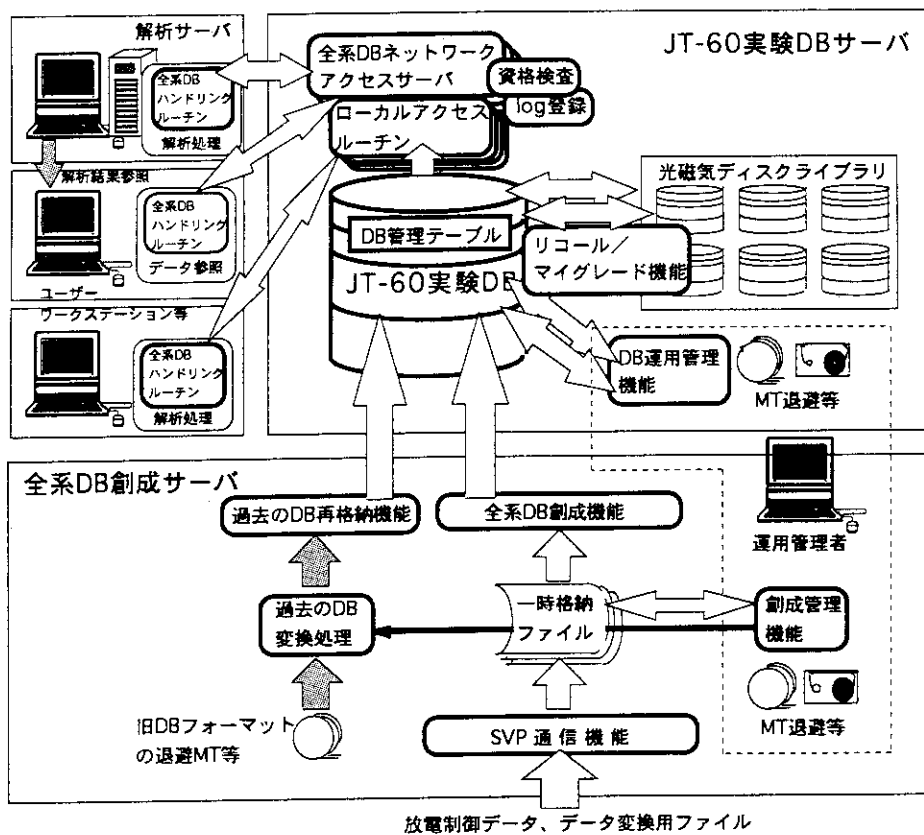
一時格納ファイルに対する運用管理機能は、全系DB創成機能の異常終了のためDBサーバに全系DBを格納できなかった時等のために生データの形式でMTにデータを保管し後に再創成ができるようにするためである。

解析サーバやその他のワークステーションに作成された全系DBハンドリング・ルーチンを使ったアプリケーションプログラムは、JT-60実験DBサーバの全系DBネットワーク・アクセス・サーバに対してデータ要求を行う。全系DBネットワーク・アクセス・サーバは、要求データに対してローカルアクセスルーチンを使用して全系DBから要求対象のデータ取得して要求元に送信する。

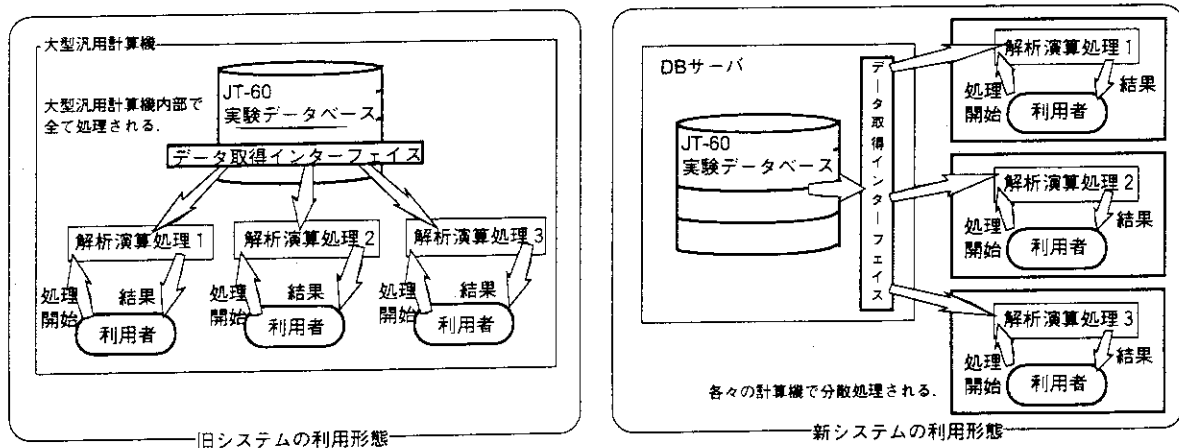
旧システムでは、データの取得、解析・結果表示が大型汎用計算機の中で全ての処理がおこなわれていた。これに対して新システムではデータの取得に関する部分のみがDBサーバの負荷となるだけで、解析・結果表示はそれぞれの計算機で処理できることで負荷の分散が図れるところに利点がある（第2.6図に旧システムと新システムでのデータ利用形態を比較して示す）。

第2.3表 各サーバのソフトウェア機能

| 計算機              | 機能名称                            | 機能   |
|------------------|---------------------------------|--|
| JT-60<br>実験DBサーバ | ネットワークアクセスサーバ<br>(ローカルアクセスルーチン) | 全系DBハンドリングルーチンからの依頼により、指定された全系DBの取得を行う。  |
|                  | リコール/マイグレード機能                   | 磁気ディスク上の全系DBを光磁気ディスクに移動する(マイグレード)機能と光磁気ディスクに退避された全系DBを必要に応じて磁気ディスクに移動(リコール)する機能。 |
|                  | 運用管理機能                          | 全系DBシステムの資源管理を行う。  |
| 全系DB<br>創成サーバ    | SVP通信機能                         | SVPで収集された放電制御データをネットワーク通信により受信する。  |
|                  | 全系DB創成機能                        | ネットワーク通信により受信した放電制御データから全系DBを創成しDBサーバに格納する。                                      |
|                  | 運用管理機能                          | SVPから受信した放電制御データ(一時格納ファイル)の資源管理を行う。  |
|                  | 過去のDB再格納機能                      | 旧システムの全系DBを新システム上に変換格納する。  |
| 解析サーバ等           | 全系DB<br>ハンドリングルーチン              | 解析サーバや一般ワークステーションに組み込み全系DBを取得する。   |



第2.5図 全系DB管理システムのソフトウェア構成



第2.6図 旧システムと新システムの利用形態

### 3. 全系DBの構成と管理の基本設計

#### 3.1 全系DBの構造検討

##### (1) 模擬システムを用いた全系DBの開発

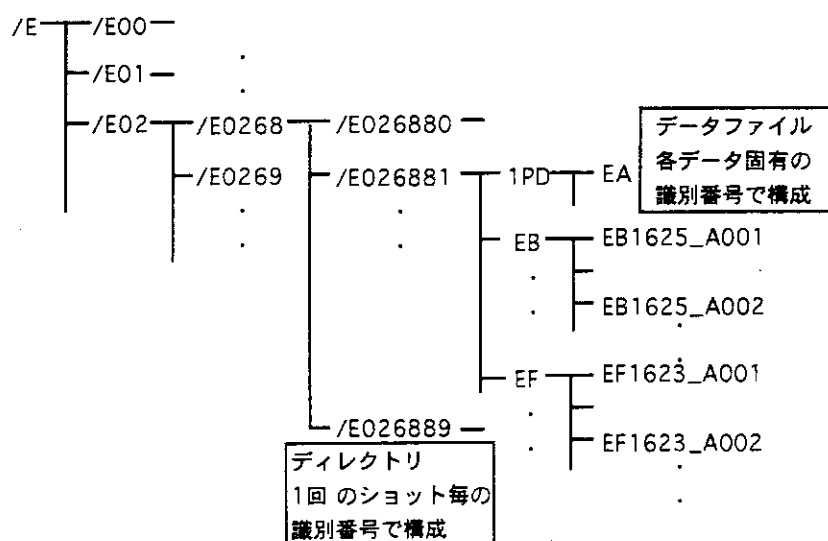
全系DBは、JT-60実験DB管理システムの開発期間に制限があったため模擬システムを用いて開発することが必要となった。これまでのJT-60実験DBの構造は階層構造でUNIXの標準ディレクトリ構造と共通点がある。そこで、このディレクトリ構造をそのまま全系DBの構造として利用することを考えた。これは、開発工数の低減を図ると言う2次効果も期待できる。しかし、この構造を用いた検索は、UNIX-OSが実行するので、必要な大量のディレクトリ分割が可能かどうか、検索の実行時間は十分短いかな等の技術的問題点の有無を確認しなければならない。

全系DBの構造は、まず最上層にショット番号と呼ばれる識別番号のツリー階層を作り、順次、細分化し最下層に個々のデータを識別するためのPID(Point IDentification)番号と呼ばれる各データ固有の番号列のデータファイルが格納される。これによりデータのアクセスは、ショット番号とPID番号から特別な検索ソフトウェアを介さずに検索出来る。

通常UNIXでは、1個のディスクパック（数100Mbyte～数Gbyte）を複数または単一のパーティションで切り分け、ユーザまたはグループ単位でディスクの使用を制限している。

現在主流の2.1GByte/1パックのディスクを最大限使える様に単一のパーティションで設定し、全系DBを格納すると1ショット当たりの全系DBの容量（約16～20MByte）から約100ショット格納可能である。

このことから第3.1図に示すようなショット番号を100ショット単位で細分化したディレクトリ階層構造を採った。



第3.1図 模擬システムを用いて開発した全系DBのディレクトリ階層構造

## (2) 実機システムに全系DBを構築する上での課題とその解決

実機として導入されたDBサーバのディスク構成は、RAID方式ディスクアレイ装置で構成された約100 Gbyteのハードディスクと約600 Gbyteの光磁気ディスク装置を接続する形で使用している。この内、全系DBの格納領域の割り当ては、磁気ディスク43 Gbyte、光磁気ディスク261 Gbyteである。磁気ディスクと光磁気ディスク間のファイル転送には、磁気ディスクの一部の領域をEpochServer<sup>\*2</sup>管理のファイルシステム（Veritas File System）として設定することにより、あらかじめ決められたしきい値に達すると自動的にアクセス日付の古いファイルから光磁気ディスクに転送（マイグレート）される。また、マイグレートされたファイルにアクセス要求があった場合、自動的に磁気ディスク上に復元（リコール）される。これにより、全系データベースのシステム設計では、ファイルが磁気ディスク、光磁気ディスクのどこにあるかを意識せずにソフトウェアの作成等ができる。また、複数のディスクパックを1つのファイルシステムとして認識させる機能を有していて、特に(1)で述べた100ショット単位のディレクトリの細分化をする必要はなかったがディレクトリの階層構造はそのままとして実機へ移行を開始した。即ち、磁気ディスク43 Gbyteと光磁気ディスク260 Gbyteの領域を1つのファイルシステムとして認識させ(1)で示したディレクトリの階層構造で全系DBを格納する方法を採れると見通しをもって設計を進めた。

更なる調査の結果、EpochServerの管理する1つのファイルシステム内に格納出来る最大ファイル数は、1,000,000ファイル以下でなければならないという制約が設計途中で明らかとなった。

全系DBの構造は、1ショット当たりのファイル数が約2,000ファイルであり、500ショット分で1,000,000ファイルとなる。これでは磁気ディスクと光磁気ディスクを有効に利用出来ないばかりか全系DBの構築が出来なくなるという問題であった。

そこで次にこの問題に対処する方法を記述する。

## ① 分割ファイルシステム

全系DBに割り当てられたの磁気ディスク及び磁気光ディスクの容量は、それぞれ磁気ディスクが43 GB、光磁気ディスクが261 GBである。これをファイルシステム数55で割りそれぞれ磁気ディスク：0.78 GB、ディスク：4.74 GBのファイルシステムを作成する。

1ファイルシステムの格納できるショット数は、

$$[5.52 \times 1000 \text{ MB}] / [16 \text{ MB}/1 \text{ shot}] = 345 \text{ shot} \text{ となり、55システム全てを実験で使うと}$$

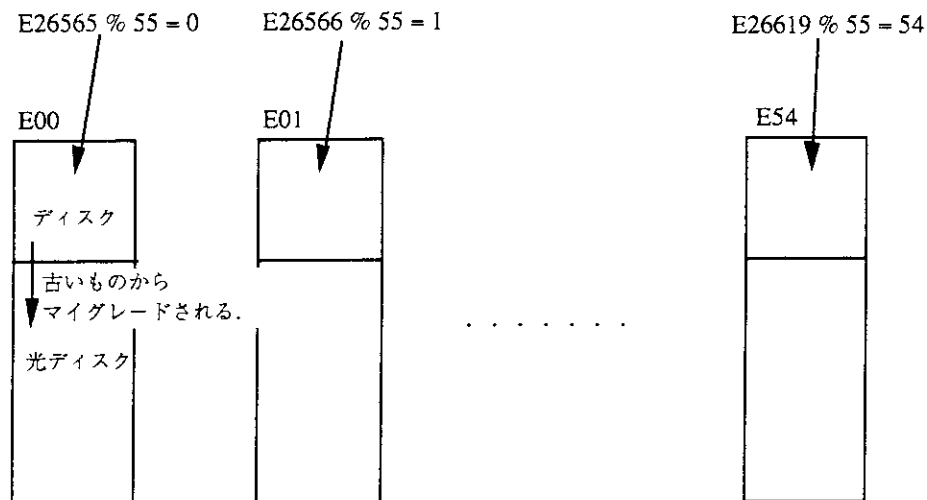
18975shotとなる。また、1ファイルシステム当たりのファイル数は、

$$345 \text{ shot} * [2000 \text{ file}/\text{shot}] = 690000 \text{ file}$$

であり、EpochServerの管理する1つのファイルシステムの許容できる範囲である。

そこでファイルシステムを効率的に運用するため、55システム全てを実験で使用したときのショットの格納方法は、ショット番号を55で割った余りの値により決定するものとした(第3.2図参照)。

\*2 EpochServer：磁気ディスク上のファイルを光磁気ディスクに移動する処理（マイグレーション）、光磁気ディスク上のファイルを磁気ディスク上に復元する処理（リコール）を自動的に行うための階層管理ソフトウェアである。



第3.2図 ファイルシステムの分割構成

これにより、ディスク上には、常に最新のショットが格納されていることになる。また、1ファイルシステムに集中的な負荷がかかることが防げる。

また、このようなファイルシステムの構成を採った場合のマイグレートとリコールに要する処理時間を調査した結果、1ショット分のデータに対応する約2,000ファイルをマイグレート/リコールするのに数十時間を要することが判明した。これは、主にファイルの作成と削除に時間を要し、処理時間は、ファイルの数に依存するものであった。一方、1ショット分と同等の容量（約20 Mbyte）の1つのファイルをマイグレート/リコールするのに要する時間は、1~2分程度である。そこで本ファイルシステムの構成を次の方法を採用することでマイグレート/リコール時間の短縮を図った。

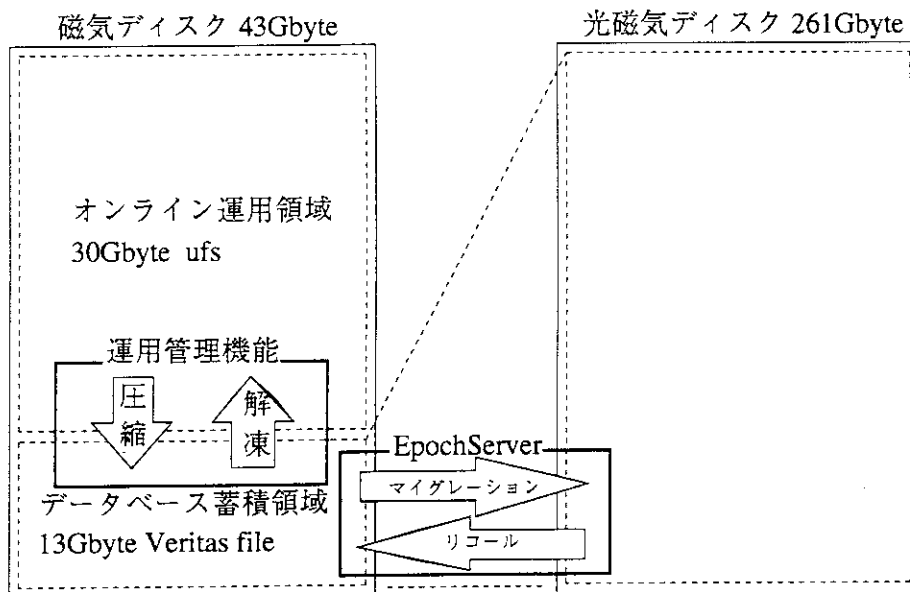
## ② 圧縮/解凍処理を追加した全系DBの構築

上の結果から、蓄積領域への格納は、1ショット当たりのファイル数を激減させなければならないことが判った。

第3.3図に示すように磁気ディスク43 Gbyteのうち30 Gbyteをオンライン領域として通常のUNIXファイルシステム（ufs：ファイル数は特に制限されない。）で管理し、残り13 Gbyteと光磁気ディスク261 Gbyteを蓄積領域としてEpochServer管理のファイルシステム（Veritas File System）で構成することとして考える。これにより、オンライン領域への全系DBの格納は、開発当初の見積もり通り、UNIXのファイル管理構成であるディレクトリ階層構造を利用し、最上層にショット番号と呼ばれる識別番号のツリー階層を作り、順次、細分化し最下層に個々のデータを識別するためのPID番号の番号列のデータファイルが格納される構造を採れる。

一方、蓄積領域であるEpochServerの管理する Veritas File System上のファイル数は、1,000,000ファイル以下でなければならないという制約があるため、全系DB蓄積領域へのファイル移動には、UNIXのtarコマンドを使用して1ショット当たり1ファイルに圧縮したファイルを作成し格納することにした。これにより、蓄積領域上のファイル数が1,000,000を越えることが防げる。





第3.3図 全系DBの格納領域の構成

### 3.2 UNIXディレクトリ構成を利用した全系DB構成

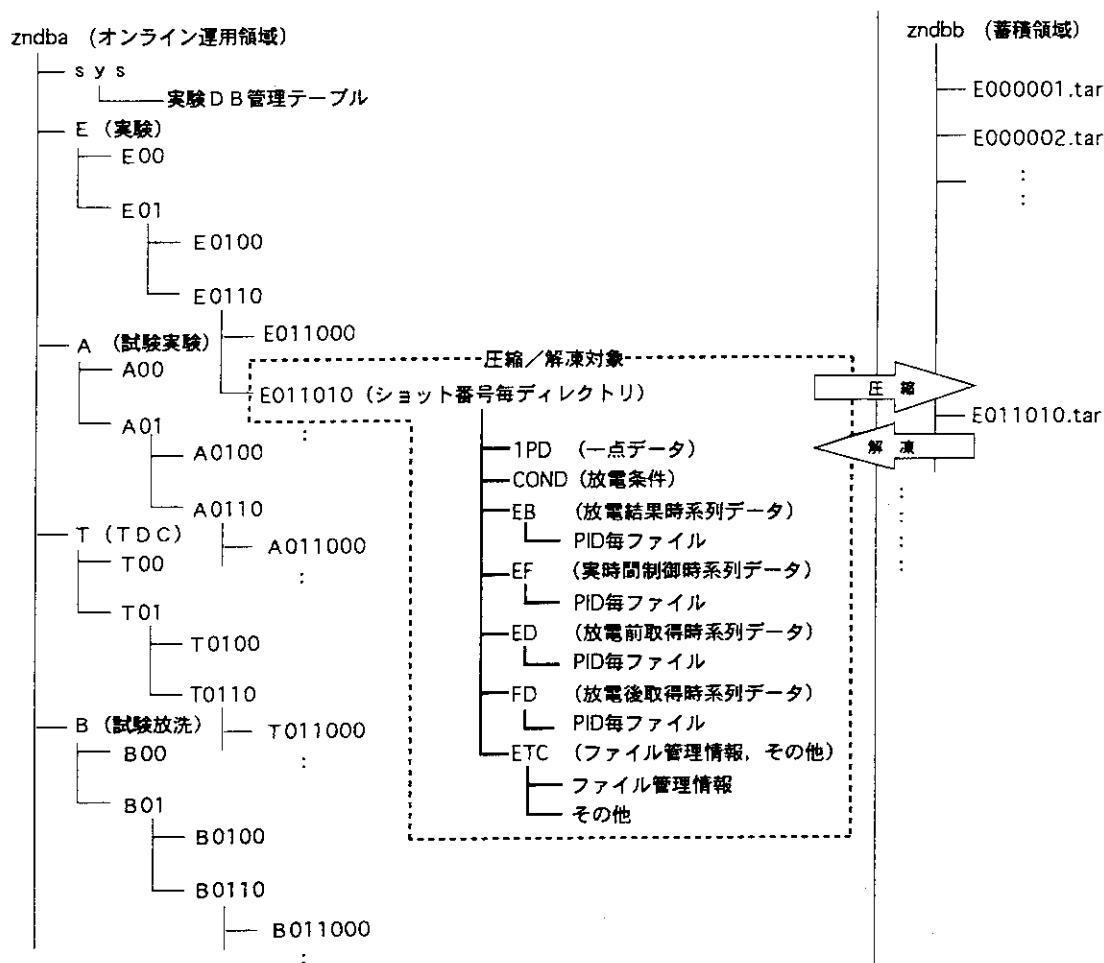
3.1節で述べたようにオンライン領域上に格納される全系DBの構造は、UNIXのファイル管理構成であるディレクトリ階層構造を利用した。まず、最上層にショット番号と呼ばれる識別番号のツリー階層を作り、順次、細分化し最下層に個々のデータを識別するためのPID番号の番号列のデータファイルが格納される。

蓄積領域へのファイル移動には、UNIXのtarコマンドを使用して1ショット当たり1ファイルに圧縮したファイルを作成し、"ショット番号.tar"というファイル名称で格納するものとした。

第3.4図中、点線で囲んだ部分が1ショット分の圧縮／解凍処理の対象データである。

これにより、オンライン領域上に格納される全系DBは、特別なデータ検索ソフトウェアを介さずショット番号とPID番号から簡単にデータの検索ができるようになる。

また、蓄積領域に圧縮されたファイルは、ユーザの利用要求に対して蓄積領域の磁気ディスク上に存在すれば解凍処理を行いオンライン領域上に展開する。この時、蓄積領域の光磁気ディスク上に存在する場合は、EpochServerが自動的に蓄積領域の磁気ディスク上に移動した後、解凍処理を行いオンライン領域上に展開する。



第3.4図 全系DBのディレクトリ構造

(1) データの圧縮/解凍処理

圧縮処理は、全系DBを蓄積領域に格納するために約2,000ファイルから構成される1ショット当たりのデータをUNIX標準のtarコマンドを使い1つのファイルに圧縮して蓄積領域に格納する機能である。解凍処理は、蓄積領域に格納されている圧縮されたファイルを利用依頼に応じて蓄積領域からオンライン領域に解凍して展開する機能である。

第3.5図に圧縮/解凍処理の概要を示す。また、圧縮/解凍に要する処理時間を第3.1表に示す。

全系DB運用管理機能からの移動要求（移動要求のタイミングについては、第6章参照。）に対して圧縮処理は、指定されたショット番号のディレクトリ以下のファイルに対してtarコマンドを発行し、蓄積領域に圧縮ファイルを作成する。

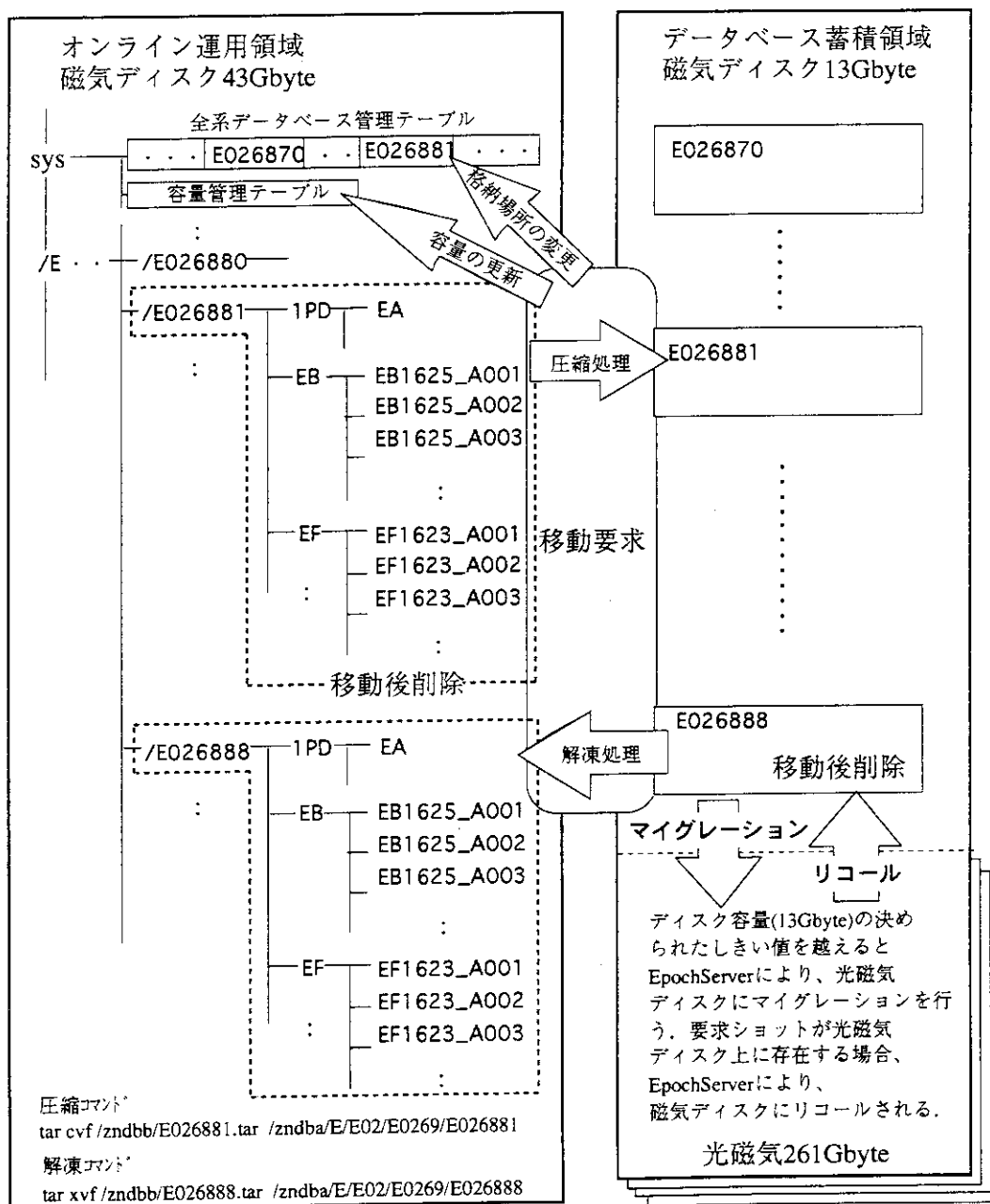
圧縮後、オンライン領域の該当ディレクトリ以下のファイルは削除する。蓄積領域に格納されている全系DBに対してアクセス要求があった場合、解凍処理は、指定されたショット番号の圧縮ファイルに対してtarコマンドを発行して圧縮ファイルを解凍しオンライン領域に展開する。

この時、指定されたショット番号の全系DBが光磁気ディスク上に格納されている場合は、EpochServerの機能により自動的に蓄積領域の磁気ディスク上に転送される。

解凍後、データベース蓄積領域の該当ファイルは削除する。

圧縮／解凍処理に伴い、全系データベース管理テーブル及び容量管理テーブルの更新を行う。

全系データベース管理テーブルと容量管理テーブルについては付録2参照。圧縮／解凍時の各テーブルの更新個所及び更新内容を第3.2表に示す。



第3.5図 解凍／圧縮処理の概要

第3.1表 圧縮／解凍に要する処理時間

|          | 圧縮処理   | 解凍処理（磁気ディスクから） | 解凍処理（光磁気ディスクから） |
|----------|--------|----------------|-----------------|
| 処理時間（平均） | 約1分20秒 | 約3分00秒*        | 約4分30秒*         |
| 処理時間（最大） | 約1分50秒 | 約4分00秒         | 約5分30秒          |

\*約1分30秒の差は、EpochServerが光磁気ディスクから磁気ディスク上の転送に要する時間である。

第3.2表 圧縮／解凍時の各テーブルの更新個所

| 更新個所               |             | 処理内容     |          |
|--------------------|-------------|----------|----------|
|                    |             | 圧縮       | 解凍       |
| 全系データベース管理<br>テーブル | DBアクセス日付    | 無し       | 解凍日付に更新  |
|                    | 格納場所        | オンライン→蓄積 | 蓄積→オンライン |
| 容量管理テーブル           | オンライン運用領域容量 | マイナス     | プラス      |
|                    | DB蓄積領域容量    | プラス      | マイナス     |

## 4. 全系DB創成機能

第3章において、全系DBの構造についての検討を行ったが、ここではそれを踏まえた上で全系DB創成に必要な機能、実際のファイル構造について説明する。

### 4.1 全系DBサーバからDBサーバへのデータ転送方法の検討

全系DBを創成する全系DB創成サーバと全系DBを格納するDBサーバがそれぞれ独立した計算機で構成された新システムでは、全系DB創成サーバ上で創成した全系DBをネットワークを介した通信により、DBサーバに転送しなければならない。

UNIXワークステーションのデータ通信（ファイル転送）においては、いくつかの方法があるが、全系DB創成の高速化と言う意味でもデータ転送に要する時間はできるだけ短いほうが良い。

第4.1表に代表的なデータ通信方法の例を示す。

第4.1表 代表的なデータ通信方法

| データ通信方法  | 機 能  |
|----------|--|
| TCP/IP通信 | Transmission Control Protocol / Internet Protocol<br>プログラムから計算機間通信を行うための機能。信頼性の保証されたプロトコル。 |
| FTP転送    | File Transfer Protocol<br>基本的にプログラムからファイル転送を行うことに利用されることを前提に設計された機能。                       |
| RCP転送    | Remote Copy<br>ユーザーが端末等からファイル転送を直接利用する機能。  |
| NFS      | Network File System<br>ネットワーク上に存在するファイルサーバのディスク資源を共有する機能。                                  |

TCT/IP通信においては、高速かつ高信頼性のデータ転送が可能となるがサーバとクライアントの双方に通信用のプログラムを作成する必要がある。

FTP,RCPにおいては、クライアント側のプログラムの作成のみで転送可能となるが一旦全系DB創成サーバ上のディスクに全系DBのすべてのファイルを作成しなければならない。ファイル転送の言葉通りサーバとクライアント双方のディスクの読み込みと書き込みが介在し転送スピードに問題があると思われた。

一方、NFSは、主メモリ上に創成された全系DBをDBサーバのディスクにあたかも自分自身のローカルディスクに全系DBファイルを作成し書き込むのと同様にプログラムの作成ができる。ディスクへの書き込みのみなので比較的転送スピードを稼げると判断した。

以上のことから全系DB創成サーバからDBサーバへのデータ転送には、NFSを使用することにした。

## 4.2 全系DB創成管理機能

全系DB運用管理者の使用するワークステーションから全系DB創成サーバ上の全系DB創成管理機能呼び出すことにより運用管理者ワークステーションの画面上に全系DB創成管理機能メニューウインドウが表示される。Xウインドウで作成されたプログラムであるため運用管理者は、Xウインドウをサポートしネットワーク上に接続されたワークステーション等であれば本機能を実行することができる。全系DB創成管理機能は、Xウインドウを利用したグラフィックインターフェイスによる操作方式を採ったため、そのほとんどの操作がマウスを使ったメニューやボタン操作で行うことができる。

全系DB創成管理機能は、次に示す全系DB創成に関する条件等を設定する。

### (1) データ変換用ファイル設定機能

SVPから転送されてくる放電制御データを格納した一時格納ファイルは、全系DB創成サーバで新システムのDBフォーマットに変換され全系DBとしてDBサーバに格納される。

この際、一時格納ファイル内のPIDデータの検索のための放電制御ミニコンピュータ上に設定された全系PID管理定数テーブルと言う情報を取得し参照する必要がある。これは、旧システムにおいても同様で旧システムでは、MTによる受け渡しであった。

本機能は、旧システムのMTによる受け渡しの煩わしさを無くするため全系DB創成管理機能画面上から起動し、ネットワークを介して簡単に取得できるようにした。

### (2) DB化不要PID選択機能

全系で収集される放電制御データの中にはデータ解析や実験運転監視には、参照する必要のないものも含まれている。DBサーバの容量確保や全系DB創成処理の高速化を図るため、不要なPIDデータをあらかじめ設定しておくことにより、設定されたPIDに関しては、全系DBとして格納しない機能を設けた。

## 4.3 全系DB創成機能

### (1) 放電制御データ変換・編集機能

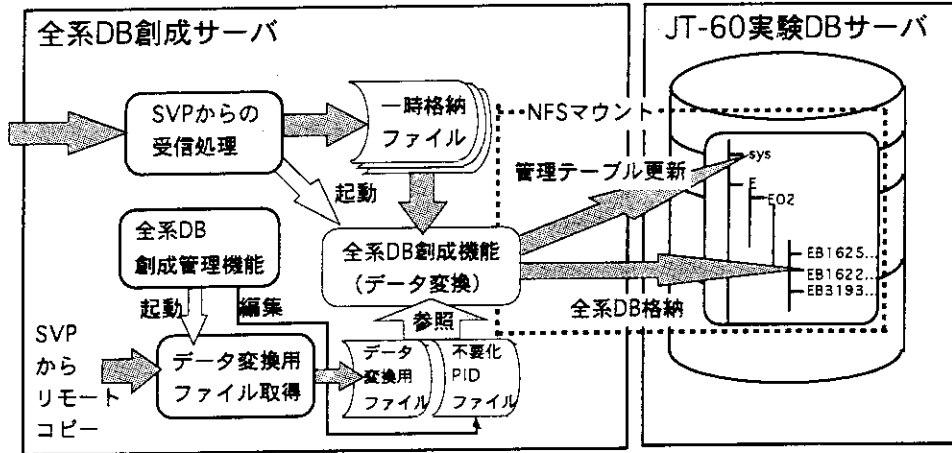
放電制御データ変換・編集機能は、SVPからの一時格納ファイル転送完了から起動され、データ変換用ファイルを基に一時格納ファイルからPID毎のデータを取得し、第4.2表に示すデータ型変換を行う。

第4.2表 全系DB創成時のデータ型変換

| 変換前 (HIDIC-80E) | 変換後 (WS)  |
|-----------------|---|
| 単精度整数 (16bit)   | 符号付き32bitの整数へ拡張                                 |
| 倍精度整数 (32bit)   | そのまま格納  |
| 実数 (32bit)      | HIDIC80E→IEEEに変換                                |
| 文字型 (8bit)      | ASCIIはそのまま格納。<br>但し、PID略称、単位は日本語<br>(EUCコード)に変換 |
| bit型            | 符号を考慮せず32bitへ拡張                                 |

変換されたデータは、全系DB創成サーバの主メモリ上で全て展開され全系DB創成サーバのディスクを使用することなくNFSによるファイル書込によりJT-60DBサーバ上のショット番号毎のディレクトリにデータ毎のファイルフォーマットで格納される。全てのデータをJT-60DBサーバに書き込むと実験DB管理ファイルを更新し終了する。

第4.1図に全系DB創成機能の概要を示す。



第4.1図 全系DB創成機能の概要

(2) ファイル構造

時系列データ及びプレプロデータについては、1PID毎のデータ点数の多いものについては基本設計通り1PID/1fileの構成を採り、1点データや放電条件の様な1PID毎のデータ量の少ないものについてはファイル作成に対する時間の増加を避けるためデータ種別毎にまとめ複数PID/1fileの形式のファイル構造とした。仮に1点データや放電条件を1PID/1fileとした場合、150Byte程度のファイルが2,000ファイル以上増えることになる。ディスクへの書き込みで一番効率の悪い書き込み方となり、ファイルの作成に非常に時間を要することになる。圧縮/解凍処理においても時間の遅延の可能性が生じてくる。データ種別毎にまとめたことにより、データ検索は1PID/1file構成の様にはいかないがデータ種別のファイルの中を探せば良いことになる。

ディレクトリ最下層にある時系列データファイルは、ヘッダー部とデータ部から構成されヘッダー部は、日本語略称、単位、サンプリングピッチ、データ点数等が格納される。また、データ部は、全てIEEE 754規格の32bit実数に変換し格納される。

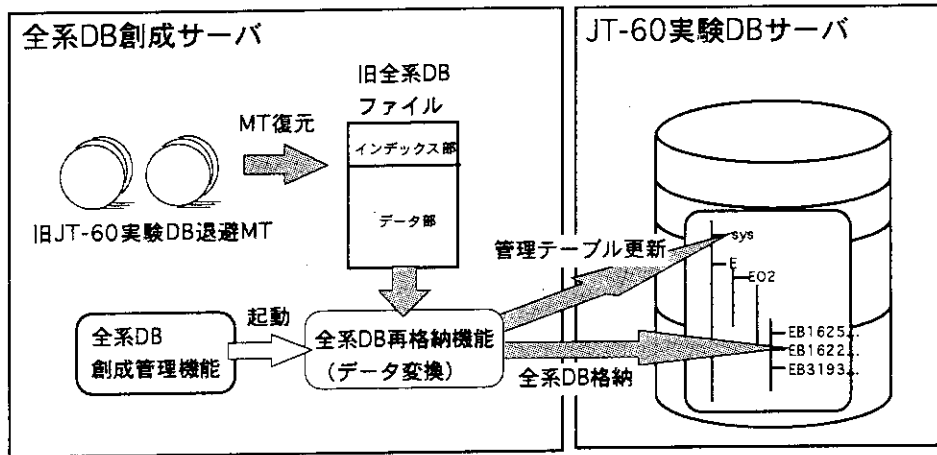
付録1に時系列データ、1点データ、放電条件及びプレプログラムデータのファイル構造を示す。

4.4 過去のデータベース再格納機能

新しいJT-60実験DBへの移行に伴う具体的な要求の内、今まで旧システムに蓄積されてきた実験DBを新システム上で利用出来ることが必要である。過去の全系DBの再格納機能は、旧システムでデータベース化されMTに退避された過去の全系DBを新しい全系DBに適合する形式にデータ変換を施した後、格納する機能である。

設計当初、開発工数の軽減を図ることを目的に全系DB創成機能と再格納機能の共通化を図れるよう再格納の基となるデータは、一時格納ファイルと呼ばれる生データを基に変換を行うこととした。ところが旧システムでデータベース化されたデータには補正係数等の誤りにより、再計算を行い格納されているデータがある。10,000ショット以上に上る再計算の必要なショットを洗いだし再計算処理を施し再格納を行うことは困難と判断し、基となるデータは、旧システムでデータベース化されMTに退避されたものを基に変換を行うこととした。

第4.2図に再格納機能の概略を示す。



第4.2図に再格納機能の概略

旧システムの全系DBは、ショット番号毎ファイル単位にMTから復元され、全系DB創成ワークステーション上で新全系DB格納方式に変換してDBサーバに格納する。

まず、MTから復元された全系DBのファイル名称からDBサーバ上にショット番号毎のディレクトリを作成する。全系DB創成サーバは、第4.3表に示すデータ型変換を行いJT-60DBサーバ上のショット番号毎のディレクトリにデータ毎のファイルフォーマットで格納される。全てのデータを変換しJT-60DBサーバに書き込むと全系DB管理ファイルを更新し終了する。付録3に時系列データを格納する場合の新全系DBファイルの変換例と1点データ、放電条件及びプレプログラムデータの旧全系DBと新全系DBのフォーマットを示す。

第4.3表 再格納時のデータ型変換

| 変換前 (FEP)     | 変換後 (WS)  |
|---------------|---|
| 倍精度整数 (32bit) | そのまま格納  |
| 実数 (32bit)    | IBM→IEEEに変換 *1                                    |
| 文字型 (8bit)    | EBCDIC→ASCIIに変換<br>但し、PID略称、単位は日本語<br>(EUCコード)に変換 |
| bit型 (32bit)  | そのまま格納  |

\*1 IBM,IEEEフォーマットについては、付録4参照。



## 5. ネットワークを介したデータベースへのアクセス

新旧のJT-60実験DB管理システムの大きな相違点は、旧システムでは、データの取得及びその解析処理等が1台の大型計算機で集中して行なわれていたのに対して新システムでは、データベースの管理計算機と解析処理を行う計算機が別の計算機で分散して行なわれることである。このためデータベースからのデータの取得には、計算機間のデータ通信が必要となる。

一方、ユーザ側から見ると旧システムのソフトウェア資源を出来るだけ手を加えずに新システムに移行できることが効率的であると共に計算機間のデータ通信のための煩雑な手続き等は、極力避けたいというユーザからの要望があった。

そこで新システムでの全系DBの取得においては、

- ① 旧システムのFORTRAN言語で作成された汎用ハンドリングパッケージと同一のサブルーチン名及び引数並びでデータの取得ができること。これによりユーザは、全系DBハンドリングルーチンを利用することによりネットワークを全く意識せずにデータの取得ができる。
- ② 特定の計算機だけではなくネットワーク上に接続された様々な機種ワークステーション等からデータの取得ができること。

以上を基本にネットワークを介したデータベースへのアクセスのためのソフトウェアを開発した。

ネットワークを介したデータベースへのアクセス機能は、ネットワーク上に接続されたワークステーション等からDBサーバの全系DBにアクセスし、全系DBに格納された任意のデータを取得するための機能である。

この機能は、DBサーバ上で動作するローカル・アクセス・ルーチンとネットワーク・アクセス・サーバ並びに一般ワークステーション上に組み込む全系DBハンドリング・パッケージから構成される。第5.1図にネットワークを介したデータベースへのアクセス機能の動作概要を示す。

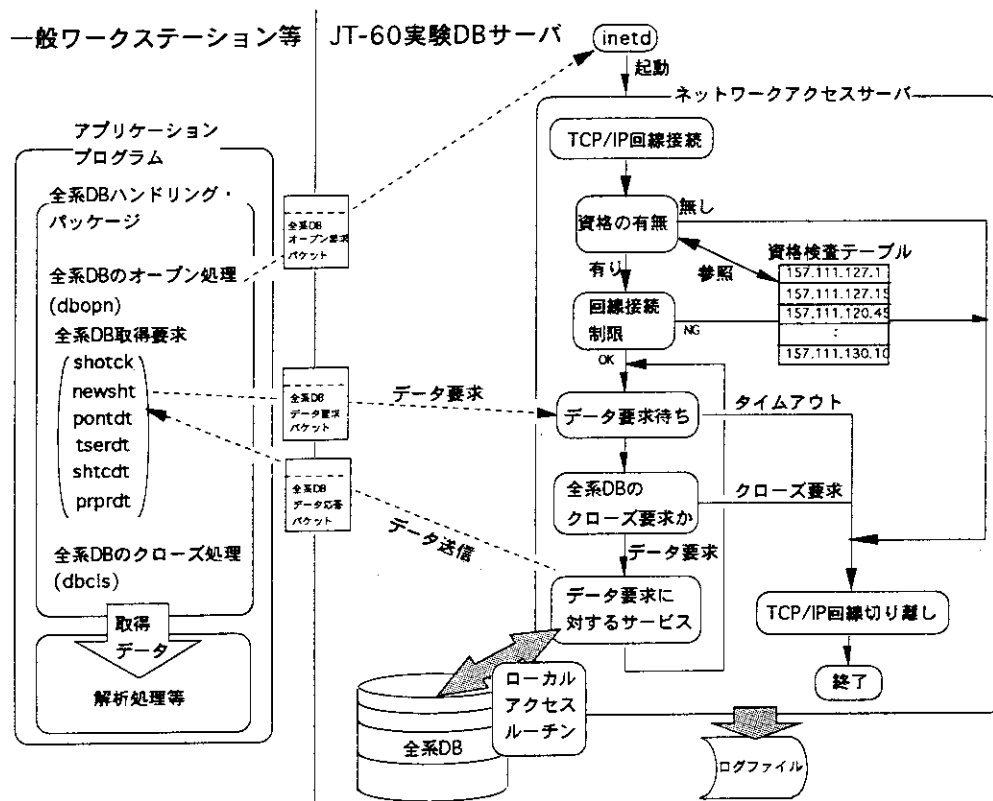
一般ワークステーション等からの全系DBオープン処理 (dbopn) 要求に対して、DBサーバは、UNIXの標準機能であるインターネット・サーバ (inetd<sup>\*3</sup>) を介してネットワーク・アクセス・サーバを起動する。ネットワーク・アクセス・サーバは、インターネット・サーバから引き渡されたTCP/IPポートと通信回線を接続する。

ネットワーク・アクセス・サーバと全系DBハンドリング・パッケージは、送受信方向と要求種別、データ部の長さ及びデータ部からなる通信パケット (付録6.1図～付録6.6図 通信パケットのフォーマット参照) でデータの取り合いをおこなう。要求元アプリケーションプログラムは、従来の大型計算機と同様にショット番号とPID番号をキーとして全系DBハンドリング・パッケージのサブルーチンを呼び出すことにより、希望するデータを取得できる。

全系DBオープン処理により通信回線接続後、ネットワーク・アクセス・サーバは、あらかじめ設定された資格検査テーブルと要求元のIPアドレスを比較して資格があるワークステーション等であ

\*3 inetd : 計算機システム立ち上げ時に起動され常時動作しているプロセスである。TCP/IPプロトコルを用いた計算機間通信を行う場合、通信ポート番号と通信プロセスをあらかじめシステムファイルに設定しておけば外部からの接続要求に対してinetdが自動的に接続用ソケットを作成し、通信プロセスにソケットを引き渡す。

れば接続とする。また、同一のIPアドレスを持つワークステーションからの回線接続数を検査し、制限値以内であれば接続とする。これらの検査に一致しない場合、通信回線を切り離し終了する。資格検査と回線接続数の検査後、次のデータ要求を受信するまで待機する。次の要求が一定時間無い場合、タイムアウトとして回線を切り離し終了する。データ要求を受信した場合、処理種別を判別し全系DBのクローズ要求であれば回線を切り離し終了する。全系DBのデータ要求の場合、ローカル・アクセス・ルーチンを使い対象データを取得して応答パケットを作成し、要求元ワークステーションに送信する。異常発生時の対処を効率的に行えるように要求元ワークステーションとネットワーク・アクセス・サーバで間の全てのデータ要求通信は、ログファイルに書き込まれ一定期間保持こととした。



第5.1図 ネットワークを介したデータベースへのアクセス機能の動作概要

### 5.1 全系DBハンドリング・パッケージ

ユーザのアプリケーションプログラムは、ユーザ側のワークステーション等に組み込まれた全系DBハンドリング・パッケージとリンクすることにより、全系DBが利用可能となる。従って、これまでのアプリケーションプログラムは、リンクするライブラリを新パッケージに置き換えるだけでDBサーバと通信回線を接続し、DBサーバに格納された全系DBを取得することができる。

また、ハンドリング・パッケージは、C言語で作成されている。一方、これまで使用されているデータ解析等に用いられるプログラムのそのほとんどがFORTRAN言語で記述されている。C言語とFORTRAN言語では、サブルーチンの入出力形式に違いがあるためFORTRAN呼び出しのための

中間インターフェイスを用意して対応した。

新システムにおける全系DBハンドリング・パッケージで提供するサブルーチンの種類は、旧システムでの使用機能を調査の上、決定した。具体的には、大きく次の3つに分類される。

① 基本アクセスルーチン

実験データの一部または、全部を取得する際に用いるサブルーチン群である。

② 応用ルーチン

基本アクセスルーチンにて取得した各種実験データを微積分処理、補間処理、等を行うサブルーチン群である。これらの演算処理は、分散処理の基本方針に従い、全てデータ取得した要求元のワークステーション等で行うものとした。

③ データ登録アクセスルーチン

データベースの追加、修正等を行うためのサブルーチン群である。一般のユーザーは、利用出来ず、DB運用管理者のみが使用することを想定している。

全系DBハンドリングルーチンのサブルーチン一覧を第5.1表に示す。

これらのサブルーチンにより、データを取得する場合、全系データベース運用管理機能の中で利用資格（付録7資格検査テーブルの構造参照）が設定されたIPアドレスを持つワークステーション等のみに対してデータ取得が可能とした。これにより、データの不正利用を未然に防ぐことになる。

第5.1表 サブルーチン一覧表

|  | データ種別          | 名称   | 処理内容   |
|--|----------------|--|--|
| 基本<br>ア<br>ク<br>セ<br>ス<br>ル<br>ー<br>チ<br>ン | 共通ルーチン         | dbopn<br>dbcls<br>shotck<br>newsht                                 | ① 全系DBのオープン(TCP/IP回線接続)<br>② 全系DBのクローズ(TCP/IP回線切り離し)<br>③ 指定ショットの全系DB有無確認<br>④ 最新ショット番号の取得   |
|  | 一点データ          | pontdt   | ① 指定PID (一点データ) のデータ取得   |
|  | 時系列データ         | tserdt<br>tsdmax<br>tsdmin<br>tsdave                               | ① 全点または指定時間帯(ts~te)のデータ取得<br>② 全点または指定時間帯(ts~te)の最大値とその時刻<br>③ 全点または指定時間帯(ts~te)の最小値とその時刻<br>④ 全点または指定時間帯(ts~te)の平均値   |
|  | プレプログラム<br>データ | prprdt<br>prdmx<br>prdmin<br>prdave                                | ① プレプログラムデータの展開結果取得<br>② 展開結果の最大値とその時間<br>③ 展開結果の最小値とその時間<br>④ 展開結果の平均値  |
|  | 放電条件           | schdt  | ① 放電条件データの取得   |
| 応用<br>ル<br>ー<br>チ<br>ン                     | 時系列データ         | tsdint<br>tsdsbn<br>tsddif<br>tsfldt<br>tsditp<br>tsditp<br>tsnpid | ① 全点または指定時間帯(ts~te)の積分<br>② 全点または指定時間帯(ts~te)の差分<br>③ 全点または指定時間帯(ts~te)の微分<br>④ 全点または指定時間帯(ts~te)の一時遅れフィルター<br>⑤ 全点または指定時間帯(ts~te)の直線補間<br>⑥ 全点または指定時間帯(ts~te)のスプライン補間<br>⑦ 4PIDデータ全点または指定時間帯(ts~te)のデータ取得 |
| *  | 全データ形式         | pidchr<br>update   | ① 指定PIDの全データを取得<br>② 指定PIDに全データを書き込む   |

\* データ登録アクセスルーチン

## 5.2 ネットワーク・アクセス・サーバの機能

ネットワーク・アクセス・サーバは、全系DBハンドリング・パッケージを使用した全系DBの取得要求に対するデータサービスと以下に示す機能を有する。これらは、データの機密性の確保やDBサーバのメモリ資源の有効利用を図るための機能である。

### (1) 資格検査

全系DBの不正利用を未然に防ぐため全系DBのオープン処理要求受信時、DBサーバ上で動作するネットワーク・アクセス・サーバは、資格検査テーブルを参照し、要求元の計算機に参照資格または更新資格があるか検査する。

資格検査テーブルの編集は、オペレーターが予め決められたフォーマットのテキストファイルを編集し、その後、バイナリファイルに変換する。これは、TCP/IP通信機能の一部である通信回線の確立時、相手側のIPアドレスがバイナリ構造で取得できる。このため資格検査時は、バイナリ同士の比較で済むことになる。事前の変換処理により、アクセスの高速化を図るための工夫である。付録7に資格検査テーブルの構造を示す。

### (2) 接続時間のタイムアウト機能

DBサーバ上の使用メモリの無駄な増加を防ぐためタイムアウト機能を有するものとした。

ワークステーション等からの回線の接続・データ要求後、一定の時間が経過しても次のデータ要求が行なわれない場合、ワークステーション側のユーザープログラムは、使用されていないものとみなし、回線を切り離しネットワークアクセスサーバは、終了する。現在の設定値は3時間となっている。

### (3) TCP/IPソケット数の制限機能

DBサーバ上のメモリ資源確保のために一台のクライアントワークステーションからの複数の回線接続数を制限する機能を有するものとした。現在の設定値は1台あたり100回線となっている。

### (4) ログ出力機能

全系DBハンドリング・パッケージを使用した要求結果をログファイルに出力し、一定期間保持する。ログファイルには、通信時刻、処理種別、要求元IPアドレス、転送データ容量等が記録される。

## 5.3 ローカル・アクセス・ルーチンの機能

ローカル・アクセス・ルーチンは、DBサーバ上のネットワーク・アクセス・サーバに組み込まれ全系DBを読み込む。ローカル・アクセス・ルーチンは、引き数としてあたえられたショット番号及びPID番号から全系DB管理テーブル及びDBサーバ上の所定のディレクトリを検索し、データ有無の確認を行う。データの存在が確認された場合、ファイルからデータを読み込みネットワーク・アクセス・サーバに返す。存在しなければリターンコードをエラーとし終了する。付録2第2.1図に全系DB管理テーブル構成を示す。

## 6. 全系DB運用管理機能

DBサーバ上の全系DBに割り当てられた容量は、磁気ディスク43Gbyte、光磁気ディスク261Gbyteである。1ショット当たりの全系DBの容量は、約16MByteであり、磁気ディスクに約2,600ショット、光磁気ディスクに約16,000ショット格納できることになる。既にJT-60の放電実績として約28,000ショット分のデータが存在している。これらを全て新システム上に移行することは、容量上不可能であり、また、ショット数は、今後さらに増え続ける。従って外部の媒体に退避したり、それらを復元したりする機能が必要となる。

そこで、全系DBの利用や記録媒体の使用率を正確に把握し、効率良く運用管理ができるよう全系DB運用管理機能を開発した。

運用管理者の使用するワークステーションからDBサーバ上の全系DB運用管理機能呼び出すことにより運用管理者ワークステーションの画面上に全系DB運用管理機能メニューが表示される。

Xウインドウを使用して作成された運用管理画面の利点として

- ① Xウインドウをサポートするネットワーク上に接続されたワークステーション等であれば本機能を操作することができる。
- ② Xウインドウを使用したグラフィックインターフェイスによる操作方式を採ったため、そのほとんどの操作がマウスを使ったメニューやボタン操作で行うことができる。

以上のことから、旧システムのTSS端末からのキーボード操作から比べるとその操作性は、格段に向上した。

表6.1に全系DB運用管理機能一覧を示す。また、付録8に全系DB運用管理画面の一部を示す。

表6.1 全系DB運用管理機能一覧

| No.            | 機能名称                      | 機能説明  | 表示項目  |                  |                  |       |                  |  |
|----------------|---------------------------|---|---|------------------|------------------|-------|------------------|--|
| 1              | 磁気ディスクの容量使用状態表示機能         | オンライン運用領域の使用状況を把握するための情報を取得し、全系DB運用管理画面に表示する、   | ① 全容量<br>② 使用量<br>③ 空き領域の大きさ<br>④ 格納可能ショット数*  |                  |                  |       |                  |  |
| 2              | 光磁気ディスクの容量使用状態表示機能        | 蓄積領域の使用状況を把握するための情報を取得し、全系DB運用管理画面に表示する、  | ① 全容量<br>② 使用量<br>③ 空き領域の大きさ<br>④ 格納可能ショット数*  |                  |                  |       |                  |  |
| 3              | 全系DB状態表示機能                | 全系DB管理テーブルからショット番号毎のDBの有無等の状態を取得し、全系DB運用管理画面に一覧表示する、  | ① ショット番号<br>② DB創成日付<br>③ DB有無<br>④ MT退避日付<br>⑤ 退避MT番号<br>⑥ 最終アクセス日付<br>⑦ 格納場所<br>⑧ ショット毎容量 |                  |                  |       |                  |  |
| 4              | 磁気ディスクから光磁気ディスクへのファイル移動機能 | <p>オンライン運用領域に格納されている全系DBを運用管理者の操作または自動的に圧縮処理を施し、蓄積領域に移動する機能である。</p> <p>(1) 運用管理者の操作<br/>                     ショット番号を指定することにより、以下の条件に当てはまる全系DBをオンライン運用領域から蓄積領域へ移動する。<br/>                     ① ショット番号による範囲指定、② 最終アクセス日の指定<br/>                     ③ MT退避済みショット</p> <p>(2) 定刻自動移動機能<br/>                     オンライン運用領域からDB蓄積領域への移動を1日1回定刻に行う機能を設けた。本機能は全系DB運用管理画面から移動目標ショット数と最終アクセス日付の条件設定することにより、指定日以前の最終アクセス日付のショットを移動目標ショット数分選択しDB蓄積領域へ移動する。</p> <p>(3) 容量監視自動移動機能<br/>                     以下のタイミングでオンライン運用領域のショット数を監視し、あらかじめ設定されたしきい値に達すると最終アクセス日付の古い順に蓄積領域へ圧縮移動する。<br/>                     ① 全系DB創成時、② 全系DB再格納時<br/>                     ③ 全系DB復元時、④ 全系DB移動時</p> | * 空き領域の大きさ / 1ショットの容量 (20MB)で計算、  |                  |                  |       |                  |  |
| 5              | 全系DB退避機能                  | <p>実験放電で創成された全系DBは、DB創成後、ただちにMTに退避出来る機能を有し、全系DB管理ワークステーションでショット番号の一覧表を表示し、退避対象ショットを選択出来るものとした。MTに格納されるデータは、蓄積領域に格納される圧縮されたショット毎のファイルと同様として指定されたショット数分MTに退避する。</p> <p>MTフォーマット</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding: 2px;">ショット一覧<br/>ファイル</td> <td style="padding: 2px;">ショット番号<br/>毎ファイル1</td> <td style="padding: 2px;">ショット番号<br/>毎ファイル2</td> <td style="padding: 2px;">.....</td> <td style="padding: 2px;">ショット番号<br/>毎ファイルn</td> </tr> </table>  | ショット一覧<br>ファイル  | ショット番号<br>毎ファイル1 | ショット番号<br>毎ファイル2 | ..... | ショット番号<br>毎ファイルn |  |
| ショット一覧<br>ファイル | ショット番号<br>毎ファイル1          | ショット番号<br>毎ファイル2  | .....   | ショット番号<br>毎ファイルn |                  |       |                  |  |
| 6              | 全系DB選択削除機能                | 削除対象を選び出す条件を指定させ、これに従って利用頻度のチェックまたは、全系DB創成日時のチェックを行い選び出されたショットについて全系DBの削除を行う、   |   |                  |                  |       |                  |  |
| 7              | 全系DB選択復元機能                | ユーザから利用したい旨の要求があったショットの全系DBについて磁気ディスク及び光磁気ディスク上にないショットを選び出しMTから磁気ディスク上に復元するコマンドを自動的に編集し実行する。尚、復元を行う際、保持予定日を設定し予定日までは、該当ショットについては削除対象から除かれるものとした、  |   |                  |                  |       |                  |  |
| 8              | 資格検査テーブル編集機能              | 資格検査テーブル用アスキーファイルをエディタで編集し、ハトリファイルに変換する、  |   |                  |                  |       |                  |  |

## 7. 性能評価と考察

### 7.1 全系DB創成処理の実行時間

平成9年1月から運用開始した新しい実験DB管理システムと旧システムの全系DB創成処理の実行時間を測定し、比較した結果を第7.1表に示す。

実行時間の測定においては、新システムと旧システムが並行運用されていた時の実験のショットを対象とした。また、模擬システムでの開発中においても同様の測定を実施した。

第7.1表 DB創成処理時間

|      | 新システム<br>Disk Array +RAID | 模擬システム<br>一般Disk | 旧システム |
|------|---------------------------|------------------|-------|
| 処理時間 | 3分10秒                     | 2分20秒            | 1分20秒 |

測定結果から新システムでの全系DB創成処理が旧システムと比べ2分近く遅いことが判明した。模擬システムでの開発段階の試験においては、1分程の遅延であったのに対して実機システムでは、更に1分程度多くの時間を要した。

この遅延の要因として、次の2点が考えられる。

- ① NFSの環境設定の最適化が不十分。
- ② DBサーバのハードウェアの処理性能が低い。

実機システムと模擬システムのハードウェアの大きな違いは全系DBを格納するハードディスクシステムが実機システムにおいては、ディスクアレイを使用したRAIDによるデータ管理であること、模擬システムでは、通常のマルチディスクパックを使った通常のUNIX管理であることがあげられる。

そこで実機システムの全系DB創成サーバとDBサーバの双方向でNFSを使った1ショット分（約16MB、ファイル数1,400）のファイル群のコピーを行ってみた。結果は、全系DB創成サーバからDBサーバでは2分20秒、DBサーバから全系DB創成サーバでは30秒前後であり、双方の差は、1分50秒となった。また、全系DB創成サーバからDBサーバへ同等の容量の1つのファイルをコピーするのに要する時間は、10数秒である。

この結果と模擬システムの結果から、全系DB創成処理を遅延させている要因は、RAIDで管理されたディスクアレイ上での大量のファイルを作成に時間を要していることが推測できる。即ち、新システムで導入したディスクアレイのハードウェアの性能によるものの可能性が高い。

ハードウェアの高速化が望ましいが、これを改善するための1つの方法としてすぐに考えられるのが、創成された全系DBを一旦、全系DB創成サーバで保持し利用化とし、1日単位で深夜間にDBサーバに転送する。ユーザーは、その日のショットに関しては、全系DB創成サーバにアクセスすることになる。このような方式を採れば全系DB創成完了時間の短縮につながるが、一般ユーザーから全系DB創成サーバへ大量アクセスした場合には全系DB創成処理の遅延も考えられるので試験が必要である。

## 7.2 ディレクトリ階層の深さの違いによる全系DBアクセス性能

第3.1節で述べたディレクトリ階層の深さの違いによるファイル検索性能を調査した。

調査方法は、DBサーバ上でディレクトリの最上位層にあるファイルと6階層目にあるファイルに対してそれぞれ1,000,000回ファイルのオープンとクローズを実行し、その実行時間を測定した。結果は、次の通り。

- ① 最上位層にあるファイル 約25秒 (1回のオープン/クローズ：25マイクロ秒)
- ② 6階層目にあるファイル 約46秒 (1回のオープン/クローズ：46マイクロ秒)

この結果からディレクトリの深さによるファイル検索の実行時間の差は、21秒である。このことから1回当りのファイル検索の実行時間の差は、20マイクロ秒程度である。この時間は、キャッシュメモリのヒットした影響等でコアメモリ内部での速度の差と考えられる。

磁気ディスクドライブのシークスピードが10数msであることから考えると、全系DBのディレクトリ階層の違いによってファイルアクセススピード等へ大きく影響を与える場合があるとすれば、ディレクトリ自体が階層方向に分割されてディスクに格納されている場合と言える。しかし、このようなケースがどのくらいの頻度で発生するか等OSの中身の動きを解析する必要があり、運用しながら監視していくことになる。

## 7.3 ネットワークを介した全系DBのアクセス速度

新しい全系DBのネットワークを介したアクセス速度について測定を行い旧システムとのアクセス速度と比較した結果を以下の示す。

測定に使用した計算機は、第2.4図「新しいJT-60実験データベースシステムのハードウェア構成」で示された、解析サーバ、全系DB創成サーバ、一般ワークステーションを使用した。それぞれの計算機から全系DBに対して64,000byte/1PIDのデータを1,000回所得した。それぞれの取得時間と旧システムでの同様の試験結果を第7.2表に示す。

第7.2表 DB取得所要時間

|      | 解析サーバ | 全系DB創成サーバ | ワークステーション | 旧システム |
|------|-------|-----------|-----------|-------|
| 取得時間 | 1分40秒 | 2分20秒     | 3分42秒     | 1分30秒 |

JT-60実験データベースを利用するユーザは、旧システム上のソフトウェア資産のそのほとんどが解析サーバに移行したこともあり、制御棟ネットワークに接続されたワークステーションやX端末から解析サーバにloginし、全系DBを利用している。

この試験結果から解析サーバは、FEPと同程度の性能であることと全系DB創成サーバや他のワークステーションのアクセス速度が解析サーバに較べ遅いことが判った。これは、解析サーバのCPU性能が全系DB創成サーバより格段に速いこと、他のワークステーションにおいては、FDDIに直結されておらずルータが介在していることとCPU性能の差等が挙げられる。全系DBをより高速に取得しデータ解析するための手段としては、分散処理システムの利点をいかし解析サーバと同等以上の性能を持つワークステーション等をFDDIに接続して使用すればデータ解析処理延滞はJT-60実験・運転の効率は更に向上すると考えられる。



## 8. まとめ

従来の集中利用型のデータベースシステムから分散処理による新しいJT-60実験DB管理システムの開発を行い、システム性能の確認を行った。

第1章で述べたに要求と基本方針を基に開発された新しいJT-60実験DB管理システムは、以下の特徴を有する。即ち、

- ① 全系DBは、UNIX特有のディレクトリ構造をそのまま利用したショット番号からなるディレクトリ階層構造の最下層に個々のデータを識別するためのPID番号のデータファイルに格納される構造としたことにより、データ検索ソフトウェアを介さずにショット番号とPID番号から簡単にデータの検索ができる。
- ② ネットワークからの汎用アクセス機能（ハンドリング・パッケージ）を開発したことにより、データベース利用者は、任意のワークステーション等からネットワークを意識した煩雑な操作をせずに簡単にデータを取得できる。また、旧システムと同一のサブルーチンとしたことで従来の解析コード等をそのまま新システムに移行できた。
- ③ Xウインドウシステムを利用した運用管理機能は、グラフィック・インターフェイスにより運用管理の操作性を向上させた。

以上の機能のほとんどがUNIXワークステーションの標準機能を利用したことで

- ① 各機能を他のワークステーションで開発出来たことと模擬システムによるFEPとの並列運転で実機導入以前から新全系DBを蓄積出来たことにより実機導入後、スムーズな実機への移行が出来た。
- ② 開発工数の低減が図れた。

これに加えて、大型汎用計算機から比べると比較的安価なUNIXワークステーションで構成されたハードウェアシステムの採用により、コストパフォーマンスに優れたJT-60実験DBシステムの構築が出来た。

初期運用の結果判明した今後の課題として、次の点が挙げられる。

- ① 全系DB創成処理のが旧システムに較べて若干遅い点については、ハードウェアの高速化を図ることやソフトウェアの改良が望まれる。
- ② ネットワークを介して接続されるさまざまな計算機からのデータ要求があることが想像される。これらの計算機の中には、PCやMacのようなパソコン、ビット配列の異なったワークステーション、さらに64bitアーキテクチャーのワークステーション等さまざまな計算機が考えられる。よって、これらに対応できるハンドリング機能の開発が必要となってくる。

## 謝辞

今回、本JT-60実験データベースシステムの開発に際し、UNIXワークステーション及びネットワークシステム全般にわたり数々の技術的助言及び協力をしていただいた(財)高度情報科学技術研究機構 青柳 哲雄氏に心から感謝の意を表します。さらに、本作業を進めるに当たり、機能設計の検討において有益な議論及び、本報告をまとめるに当たって適切な助言をしていただいた日本原子力研究所核融合装置試験部JT-60第1試験室長 木村 豊秋氏に厚く御礼申し上げます。また、性能評価試験のデータ取得等、技術的協力をしていただいた(株)日本エクス・クロンの 清瀬 総一郎氏に心から感謝の意を表します。

## 参考文献

- [1] 栗原 研一・伊藤 康浩・木村 豊秋 JAERI-M 87-097 「JT-60全系実験データベース構築 (I) — 基本構想とシステム設計 —」 (1987年)
- [2] 伊藤 康浩・栗原 研一・木村 豊秋 JAERI-M 87-098 「JT-60全系実験データベース構築 (II) — 機能マニュアル —」 (1987年)
- [3] 村井 純・砂原 英樹・横手 靖彦 「UNIXワークステーション I」 <基礎技術編> (株)アスキー出版局 (1987年)
- [4] 上谷 晃弘 「ローカルエリアネットワーク イーサネット概説」 (株)丸善 (1989年)

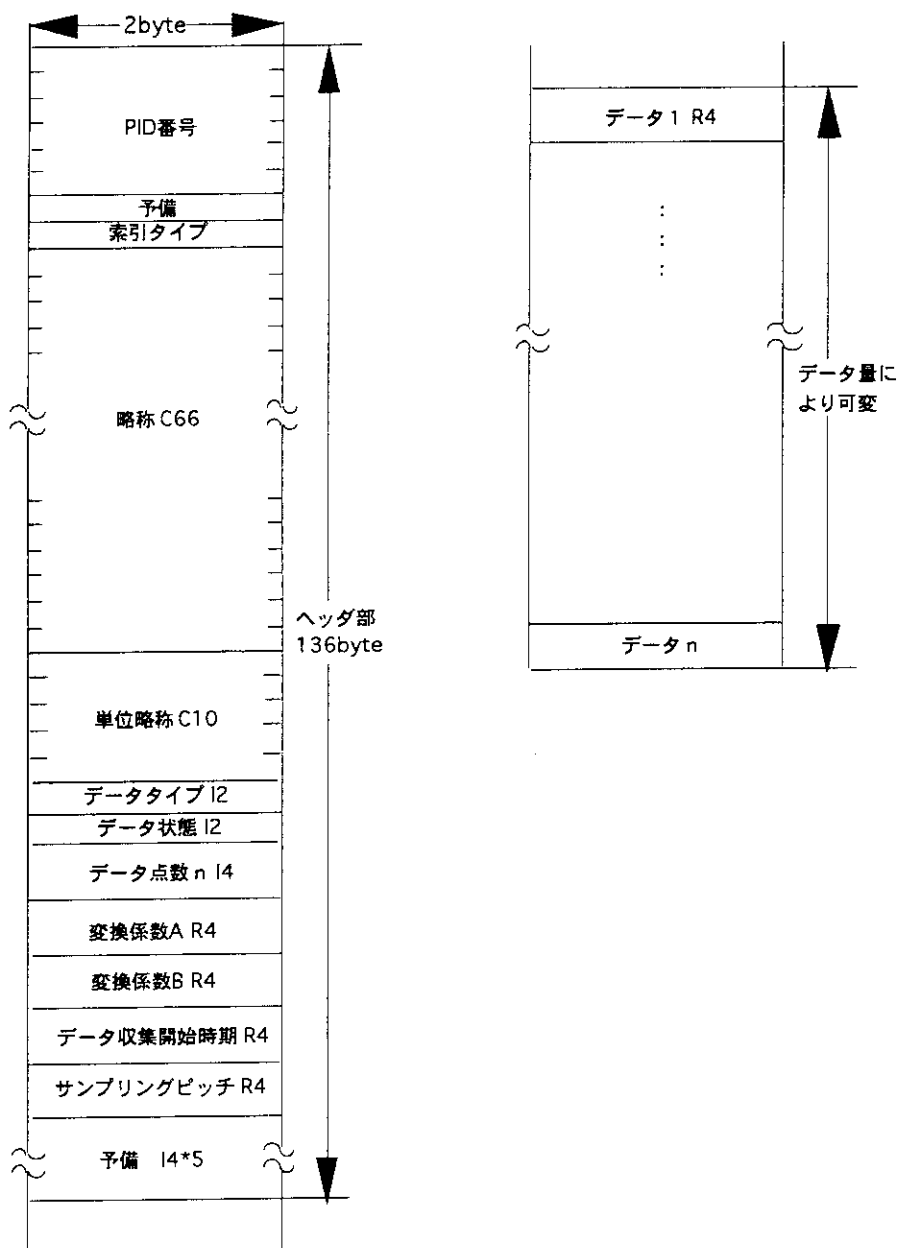
## 謝辞

今回、本JT-60実験データベースシステムの開発に際し、UNIXワークステーション及びネットワークシステム全般にわたり数々の技術的助言及び協力をしていただいた(財)高度情報科学技術研究機構 青柳 哲雄氏に心から感謝の意を表します。さらに、本作業を進めるに当たり、機能設計の検討において有益な議論及び、本報告をまとめるに当たって適切な助言をしていただいた日本原子力研究所核融合装置試験部JT-60第1試験室長 木村 豊秋氏に厚く御礼申し上げます。また、性能評価試験のデータ取得等、技術的協力をしていただいた(株)日本エクス・クロンの 清瀬 総一郎氏に心から感謝の意を表します。

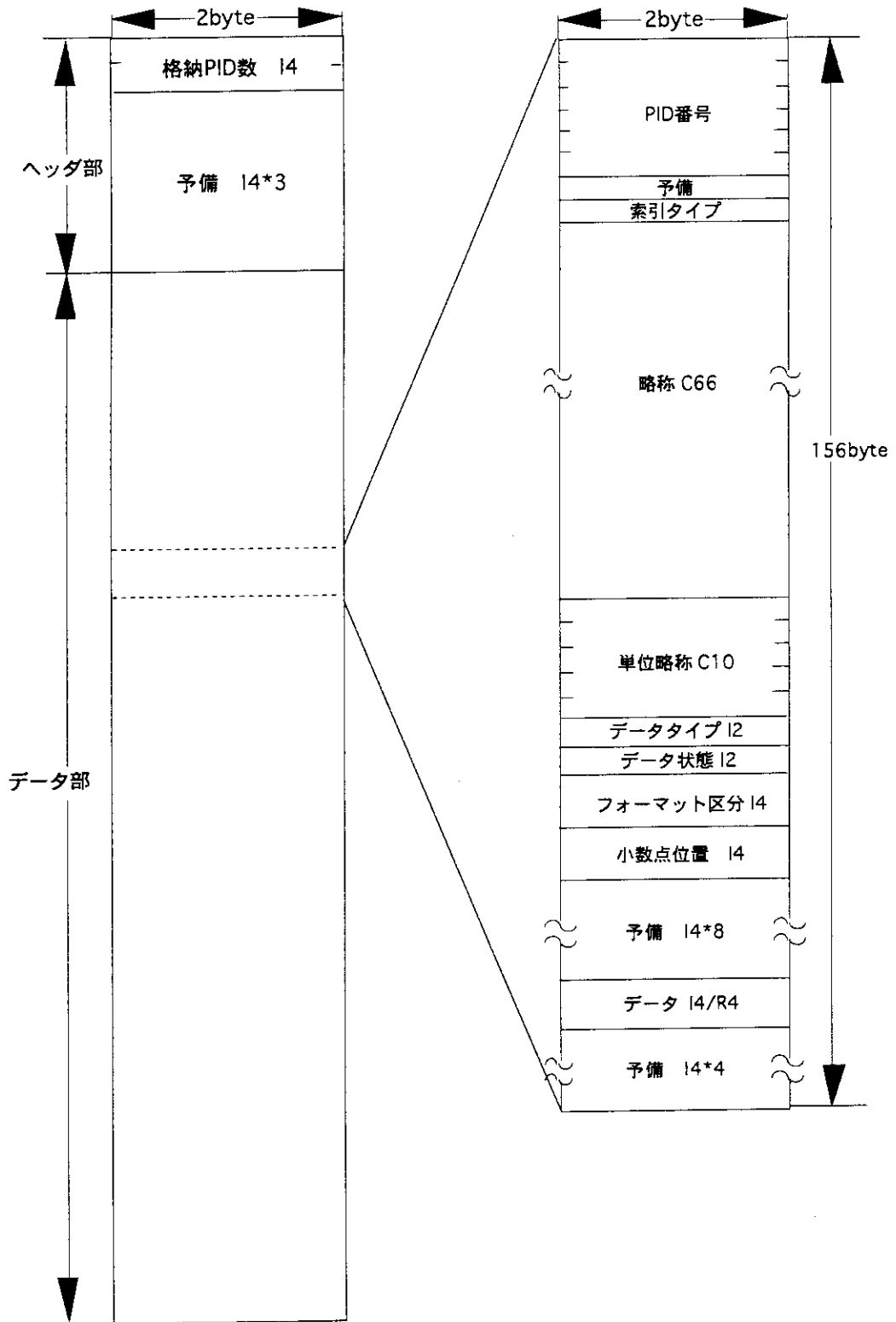
## 参考文献

- [1] 栗原 研一・伊藤 康浩・木村 豊秋 JAERI-M 87-097 「JT-60全系実験データベース構築 (I) — 基本構想とシステム設計 —」 (1987年)
- [2] 伊藤 康浩・栗原 研一・木村 豊秋 JAERI-M 87-098 「JT-60全系実験データベース構築 (II) — 機能マニュアル —」 (1987年)
- [3] 村井 純・砂原 英樹・横手 靖彦 「UNIXワークステーション I」 <基礎技術編> (株)アスキー出版局 (1987年)
- [4] 上谷 晃弘 「ローカルエリアネットワーク イーサネット概説」 (株)丸善 (1989年)

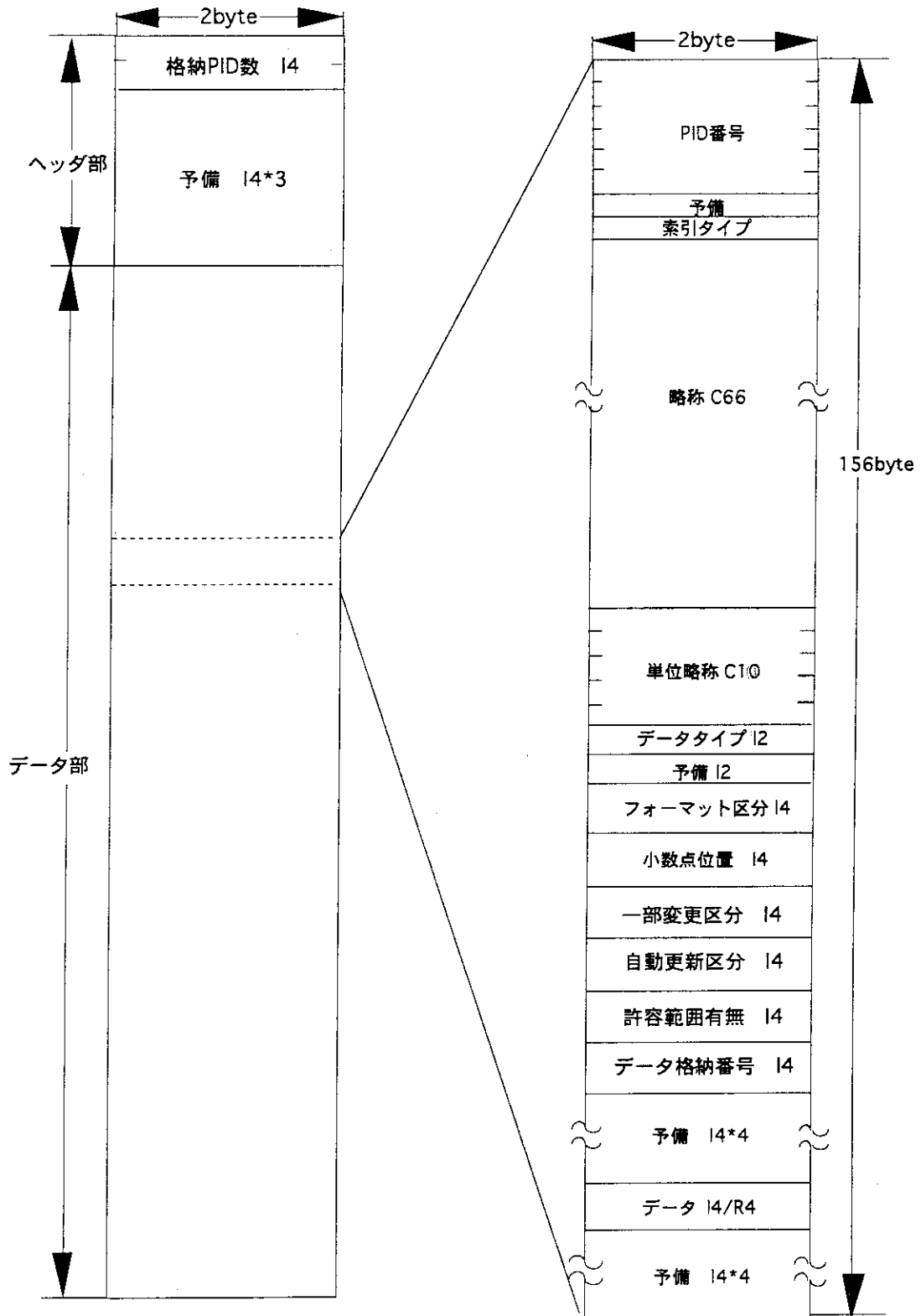
付録1 新全系DBファイルフォーマット



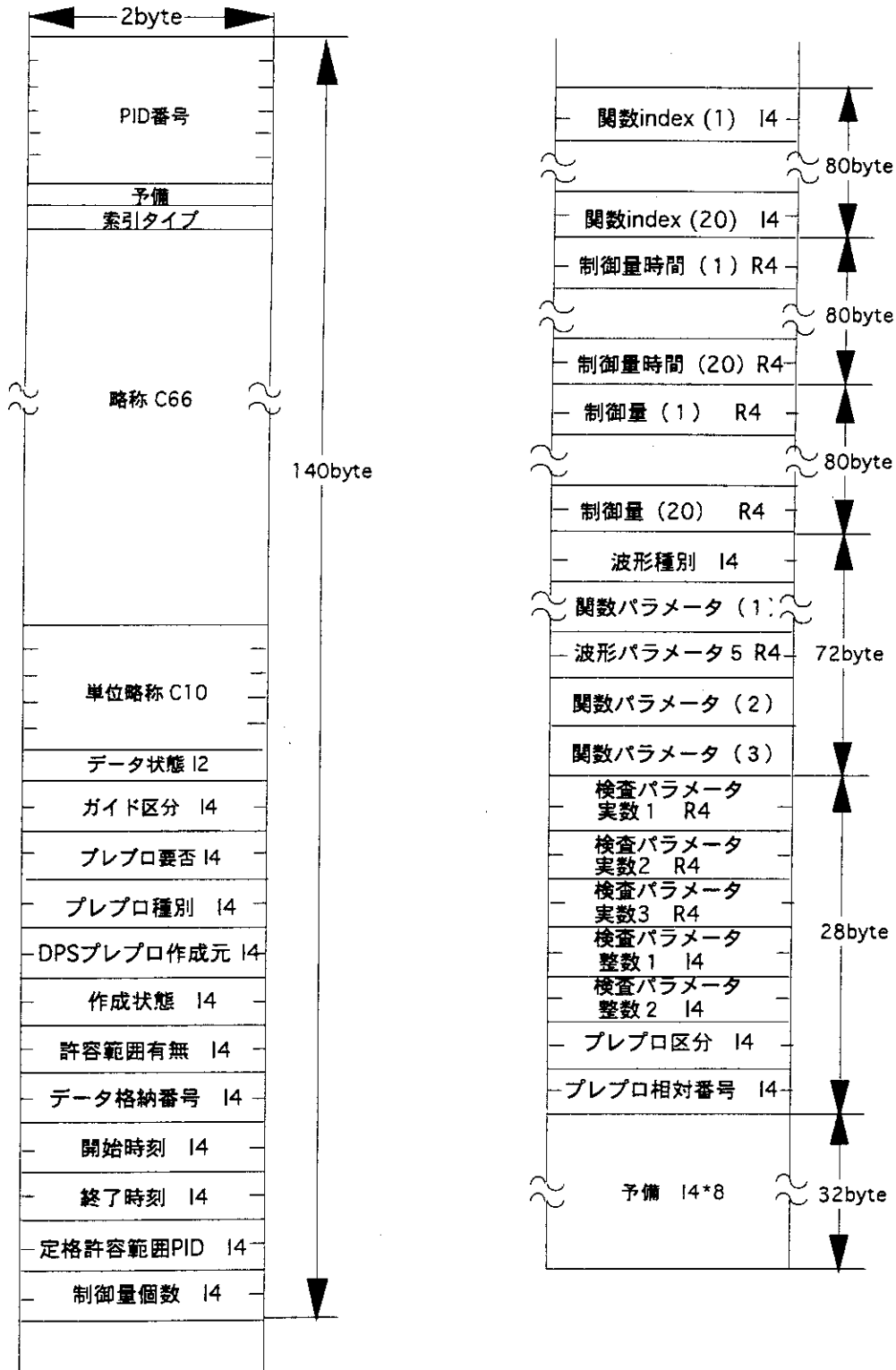
付録1.1図 時系列データのファイル構造



付録1.2図 1点データのファイル構造

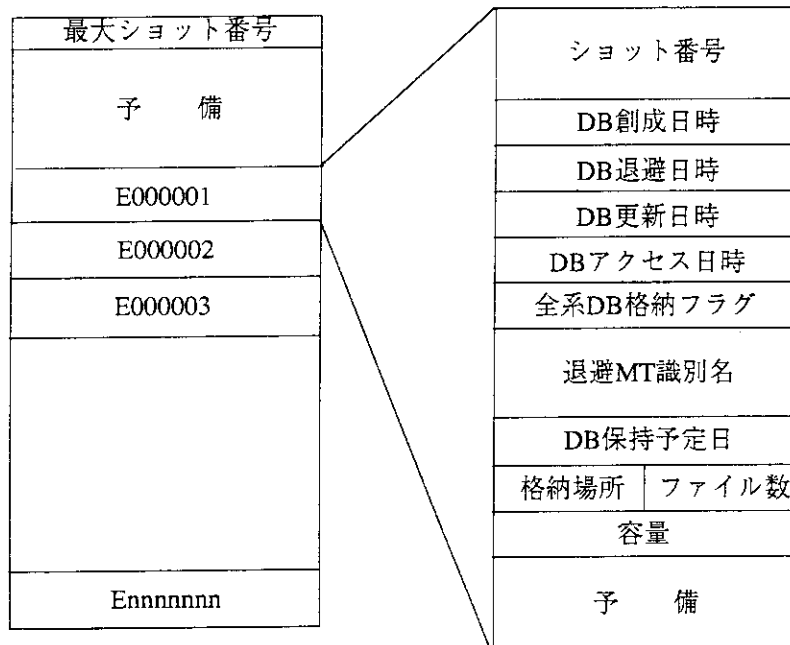


付録1.3図 放電条件のファイル構造

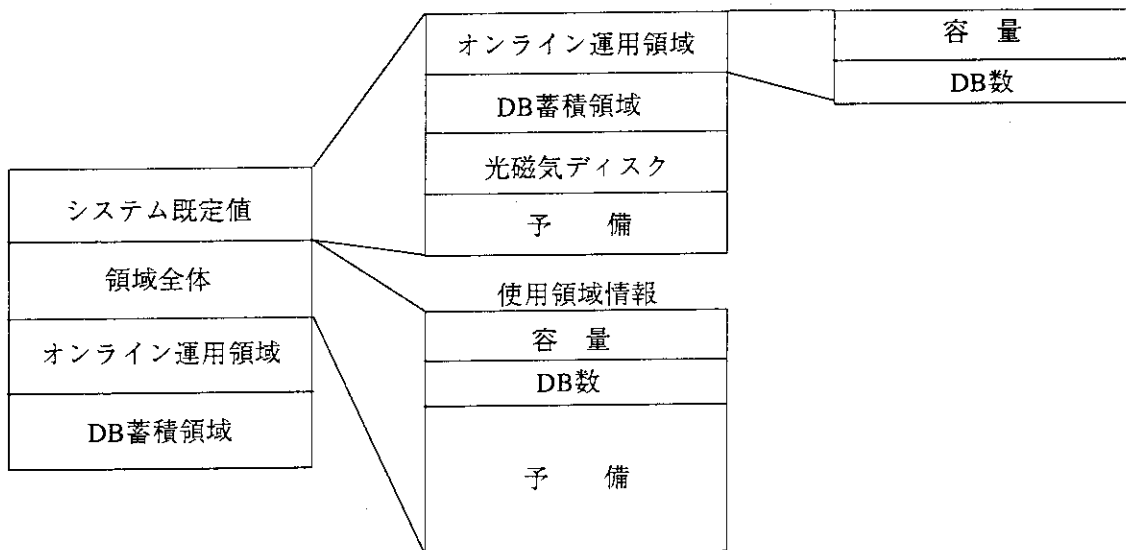


付録1.4図 プレプログラムデータのファイル構造

付録2 DB管理テーブルフォーマット



付録2.1図 全系データベース管理テーブルの構成



付録2.2図 容量管理テーブルの構成



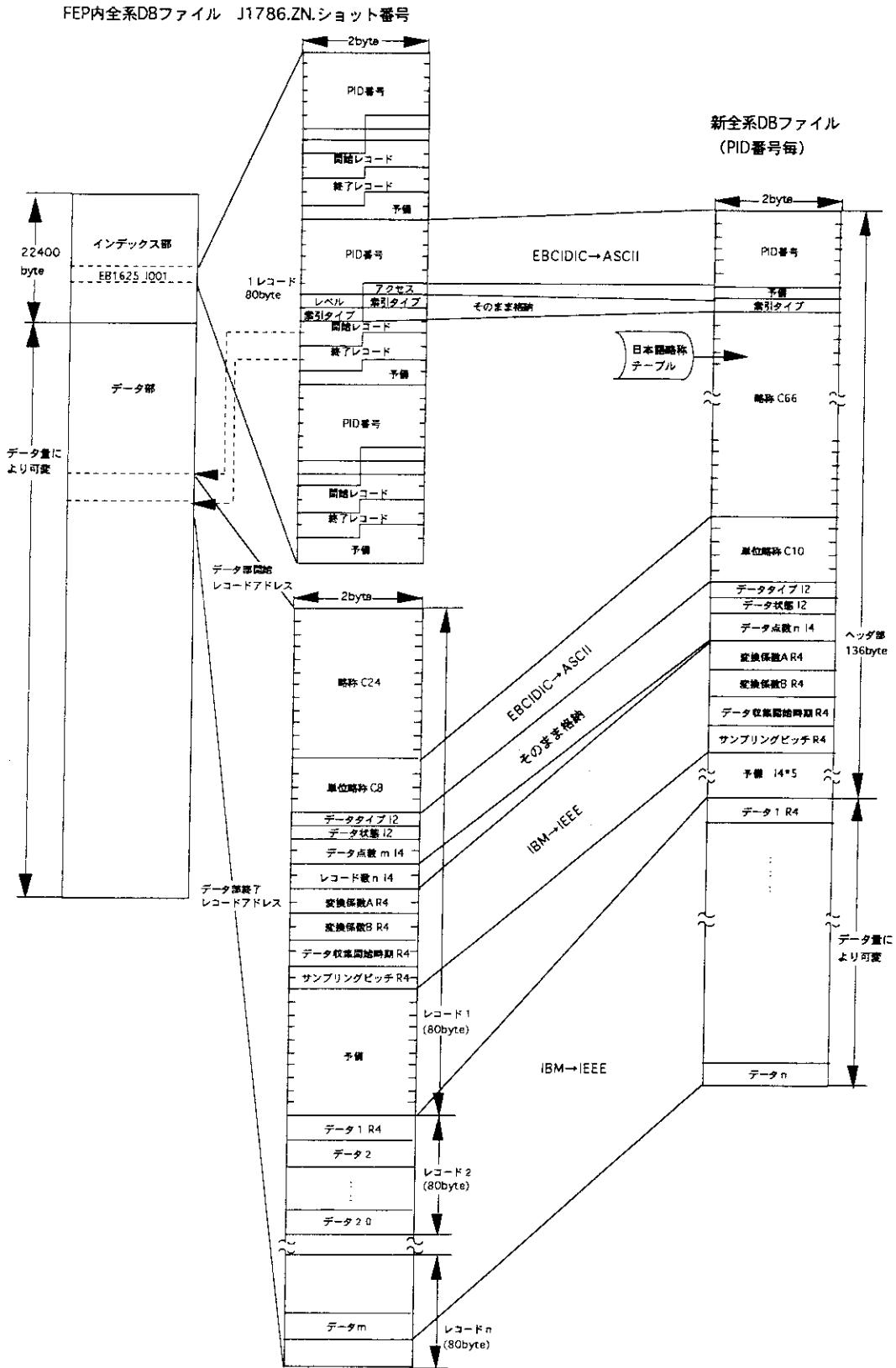
## 付録3 旧全系DBファイルから新全系DBファイルへの変換例

付録3.1図に時系列データを格納する場合の新全系DBファイルの変換例を示す。

時系列データの場合は、インデックス部とデータ部に分れた構造をとっているため以下に示す順序で変換を行う。

- ① インデックス部を1レコード(80byte)読み込む。  
1レコード中に3PID分のインデックス情報が格納されている。
- ② インデックス部の1PID分の開始レコードと終了レコードから対応するデータ部を読み込む。
- ③ インデックス部のPID番号をEBICIDICからASCIIに変換して新DBファイルの先頭からnull文字付きで格納する。
- ④ インデックス部の索引タイプを新DBファイルの15byte目からそのまま格納する。
- ⑤ データ部の単位略称をEBICIDICからASCIIコードに変換して新DBファイルの73byte目から格納する。
- ⑥ データ部のデータタイプ、データ状態、データ点数をそのまま新DBファイルの83byte目から格納する。
- ⑦ データ部の変換係数A、変換係数B、データ収集開始時期、サンプリングピッチは、実数のためIBMフォーマットからIEEEフォーマットに変換して新DBファイルの101byte目からそれぞれ格納する。
- ⑧ データ部のデータをデータ点数分IBMフォーマットからIEEEフォーマットに変換して新DBファイルの127byte目からそれぞれ格納する。
- ⑨ 日本語略称テーブルから対応するPID略称を取りだし新DBファイルの17byte目から格納する。
- ⑩ 時系列データの場合は、以上をインデックス部に格納されたPID数分繰り返す。

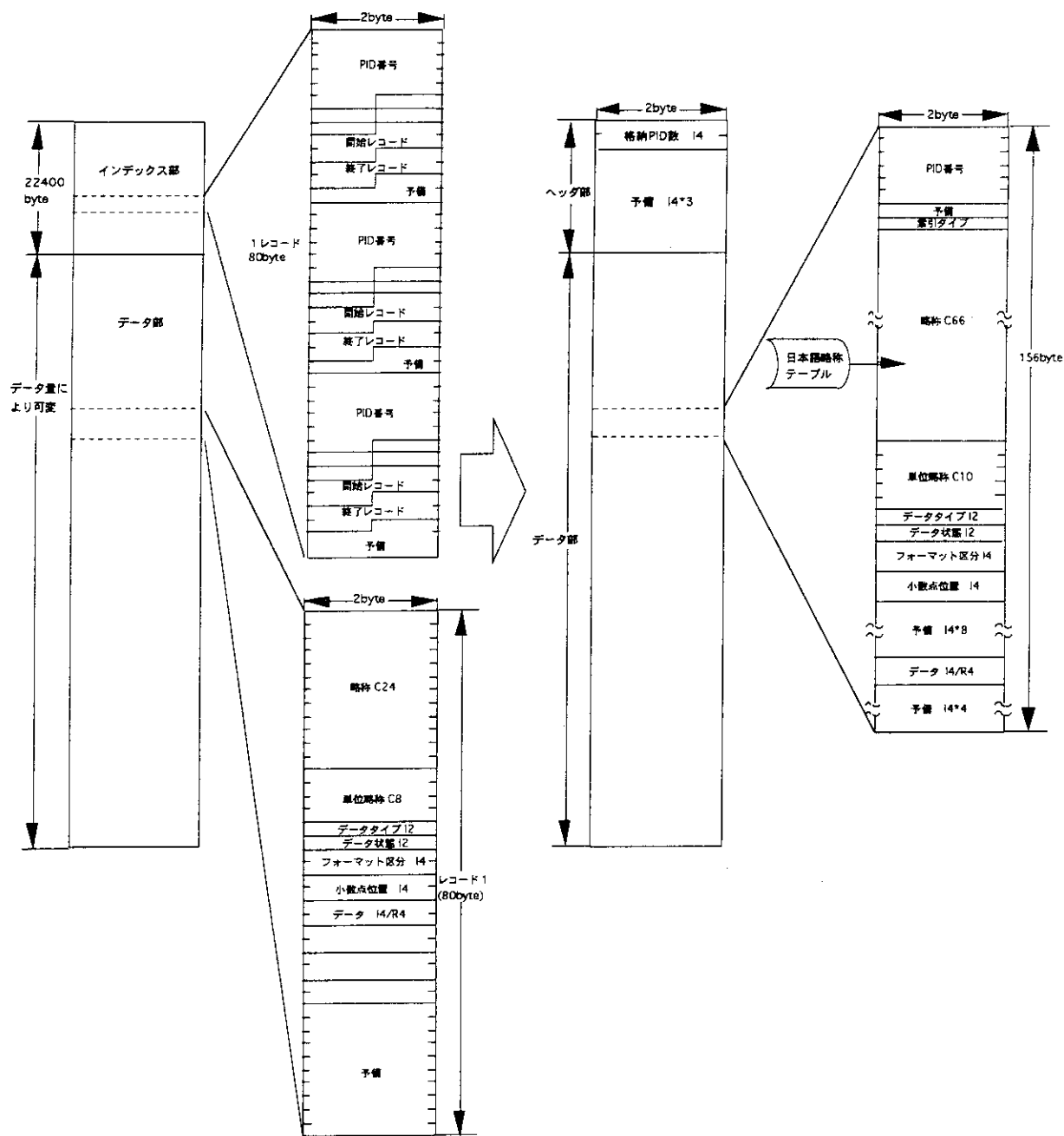
付録3.2図から付録3.4図には、それぞれ、1点データ、放電条件及びプレプログラムデータの旧全系DBと新全系DBのフォーマットを示す。



付録3.1図 時系列データの場合の新全系DBファイルの変換例

FEP内全系DBファイル J1786.ZN.ショット番号

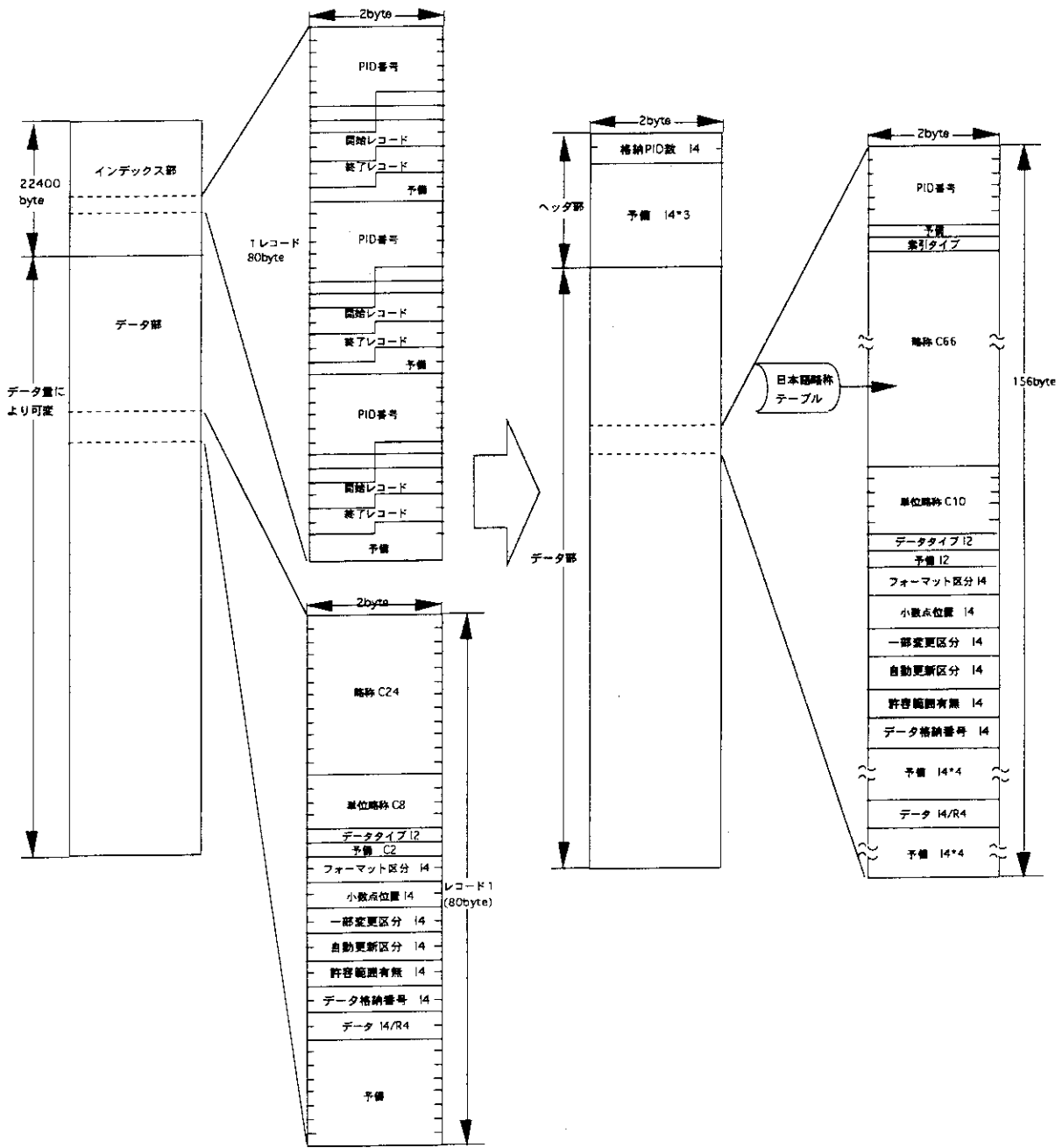
新全系DBファイル 一点データファイル



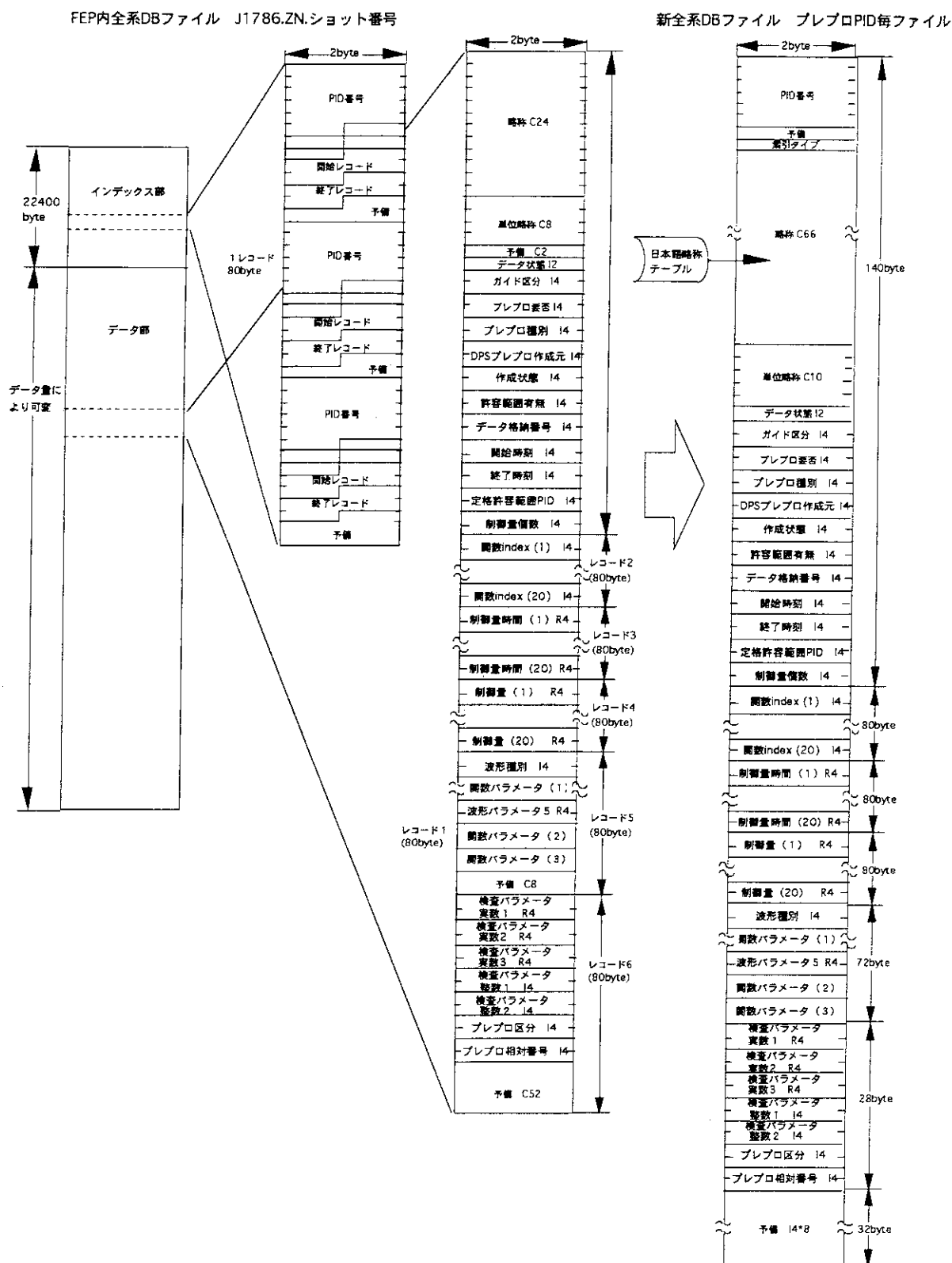
付録3.2図 一点データの旧全系DBフォーマットと新全系DBフォーマット

FEP内全系DBファイル J1786.ZN.ショット番号

新全系DBファイル 放電条件ファイル



付録3.3図 放電条件ファイル旧全系DBフォーマットと新全系DBフォーマット



付録3.4図 プレプロの旧全系DBフォーマットと新全系DBフォーマット

付録4 IBMフォーマットとIEEEフォーマットのデータ表現形式

IBMフォーマットとIEEEフォーマットの実数データ表現形式を以下に示す。

(1) IBMフォーマット

|    |             |   |   |    |   |   |   |      |   |   |   |    |   |   |   |    |  |  |  |
|----|-------------|---|---|----|---|---|---|------|---|---|---|----|---|---|---|----|--|--|--|
| 0  | 1           | 2 | 3 | 4  | 5 | 6 | 7 | 8    | 9 | A | B | C  | D | E | F |    |  |  |  |
| S  | 指数部E (7bit) |   |   |    |   |   |   | m1   |   |   |   | m2 |   |   |   |    |  |  |  |
| m3 |             |   |   | m4 |   |   |   | 仮数部M |   |   |   | m5 |   |   |   | m6 |  |  |  |

S : 符号 (= 0 : 正または0、= 1 : 負)

E : 指数 ( $0 < E < 127$ )

M : 仮数 ( $16^{-1} \leq M \leq 1 - 16^{-7}$ )  $M = m1 \times 16^{-1} + m2 \times 16^{-2} \dots m6 \times 16^{-6}$

数値 =  $(-1)^S \times M \times 16^{E-64}$

(2) IEEEフォーマット

|      |             |   |   |   |   |   |   |   |    |       |   |   |   |   |   |     |
|------|-------------|---|---|---|---|---|---|---|----|-------|---|---|---|---|---|-----|
| 0    | 1           | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9  | A     | B | C | D | E | F |     |
| S    | 指数部E (8bit) |   |   |   |   |   |   |   | m1 | ..... |   |   |   |   |   |     |
| 仮数部M |             |   |   |   |   |   |   |   |    | ..... |   |   |   |   |   | m23 |

S : 符号 (= 0 : 正または0、= 1 : 負)

E : 指数 ( $0 < E < 255$ )

M : 仮数 ( $2^{-1} \leq M \leq 1 - 2^{-24}$ )  $M = m1 \times 2^{-1} + m2 \times 2^{-2} \dots m23 \times 2^{-23}$

数値 =  $(-1)^S \times (1.0 + M) \times 2^{E-127}$

付録5 全系DBハンドリング・パッケージの問題点

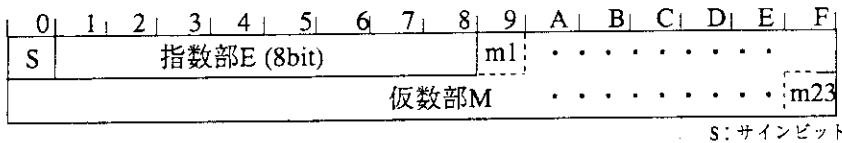
全系DBハンドリング・パッケージを利用し、接続を行う計算機の中には、DBサーバとメモリ表現形式の異なるものや、コンパイル環境の違いにより、正常に動作しないものがある。

(1) データ表現形式の違い

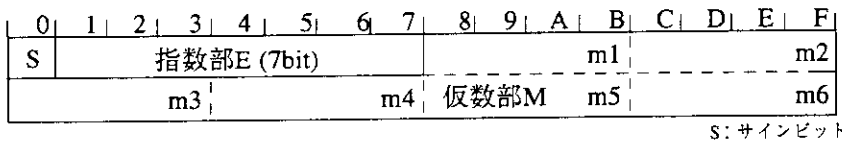
① オペレーティングシステムの違いによる浮動小数点表現形式の違い。

IEEEフォーマットやIBMフォーマットの違い。

・ IEEEフォーマット



・ IBMフォーマット

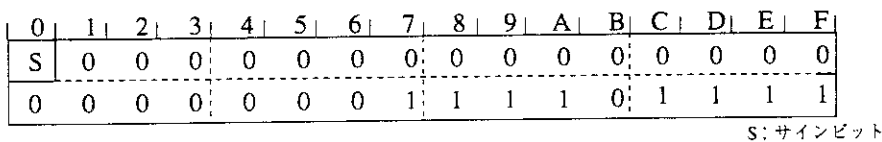


② CPUアーキテクチャの違いによりバイト順序が逆転している。

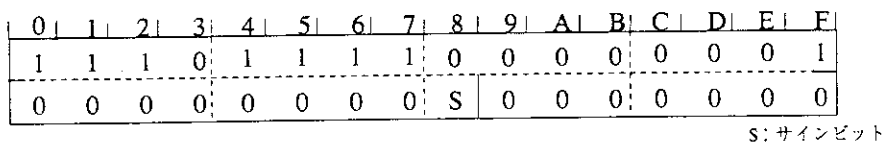
Motorola 680x0系 (BigEndan) とIntel 80x86系 (LITTLEEndan) の違い。

4byte整数+495をそれぞれの形式で現すと、

・ Motorola 680x0系 (BigEndan)



・ Intel 80x86系 (LITTLEEndan)



となる。(浮動小数点表現形式でも同様)

Motorola 680x0系で+495を現したデータをIntel系で取り込むと-32505856を現すことになる。よって、全系DB (Motorola 680x0系) をIntel 80x86系CPU搭載のワークステーション等で取得する場合、2バイト以上で構成されるデータは、バイトスワップ処理が必要となる。

(2) コンパイル環境の違い

C-言語で記述された全系DBハンドリング・パッケージをFORTRANから利用する場合の呼びだ

し形式がコンパイラにより異なっている。

例として、Sun Solaris系とIBM System-V系の違いを述べる。

① Sun Solaris系の場合

C-言語で記述されたサブルーチンをFORTRANでコールする場合、サブルーチン側の名称の最後に "\_" を付ける必要がある。

② IBM System-V系

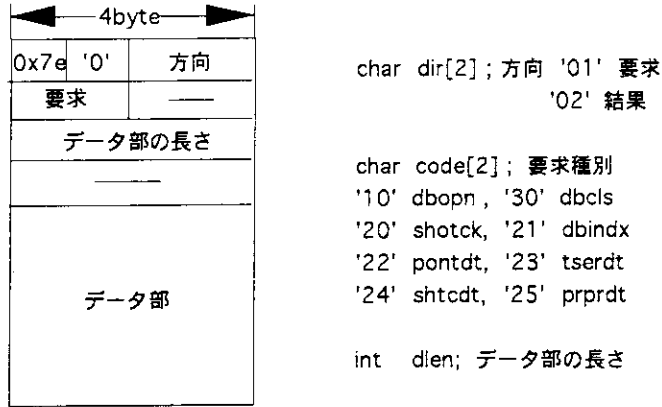
C-言語で記述されたサブルーチン名をFORTRANから直接コールする。

この問題点については、それぞれのコンパイラを調査し、それに合わせたプログラム修正が必要となる。

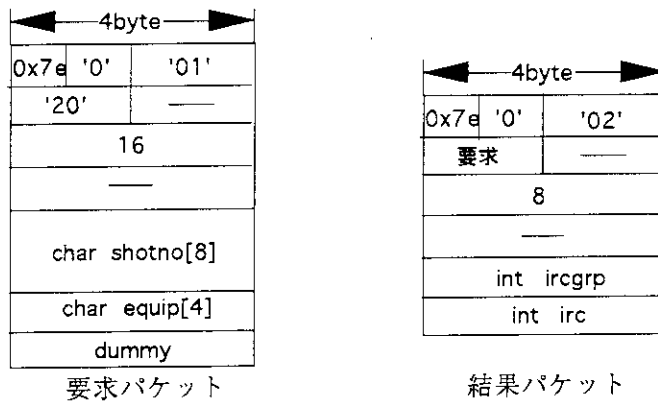
全系DBハンドリング・パッケージでは、コンパイル時のフラグの切換により、Sun Solaris系とIBM System-V系の2種類に対応出来ることとした。



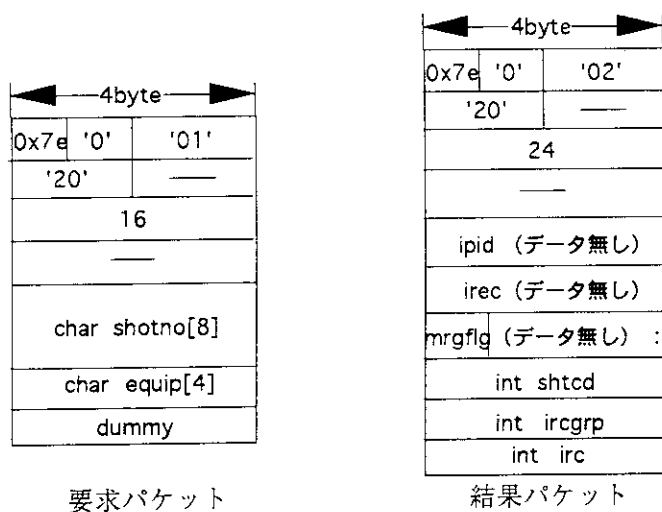
付録6 基本サブルーチン通信フォーマット



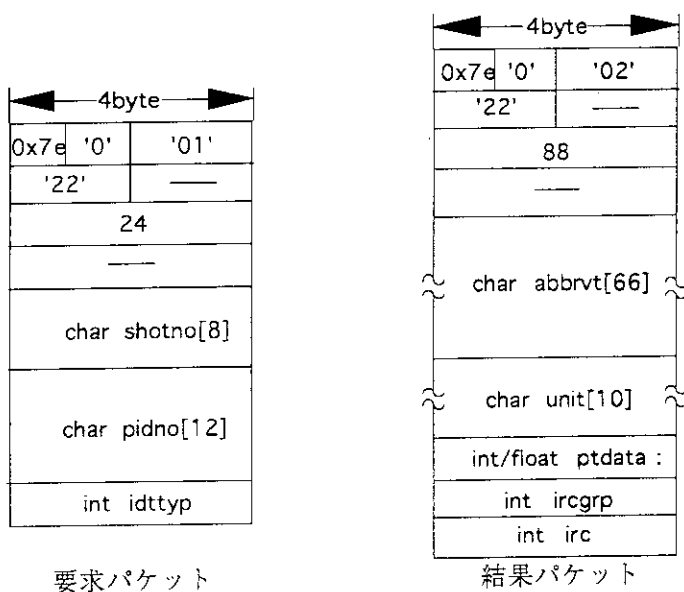
付録6.1図 標準パケットフォーマット



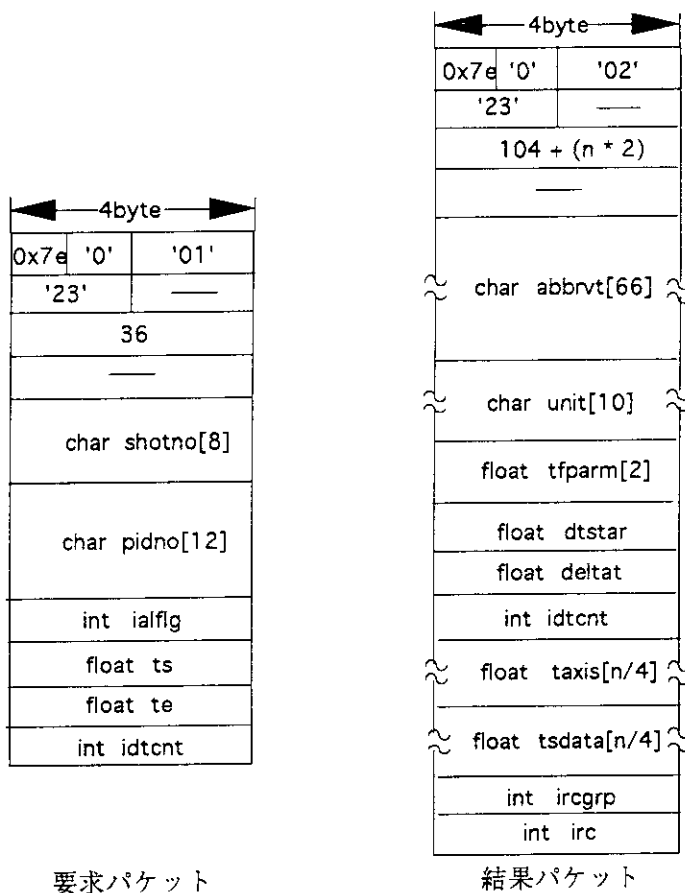
付録6.1図 dbopn, dbclsパケットフォーマット



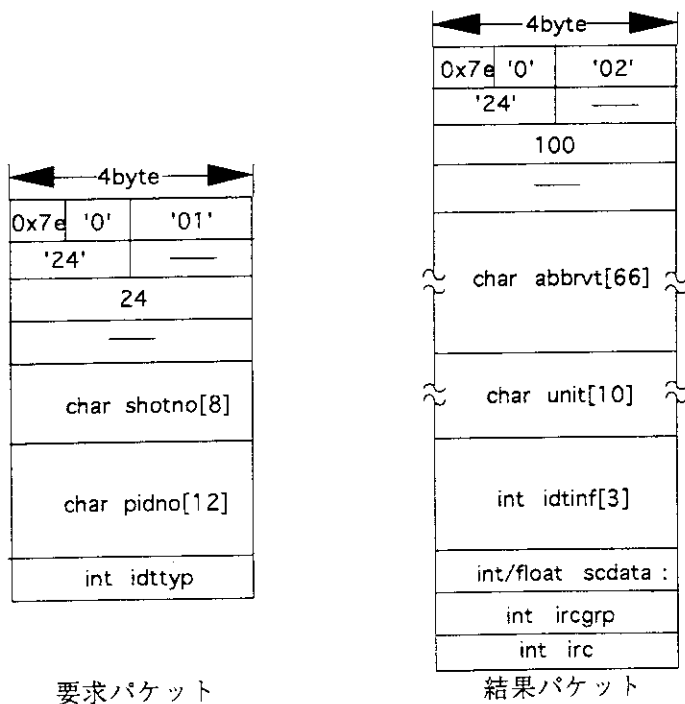
付録6.2図 shotck パケットフォーマット



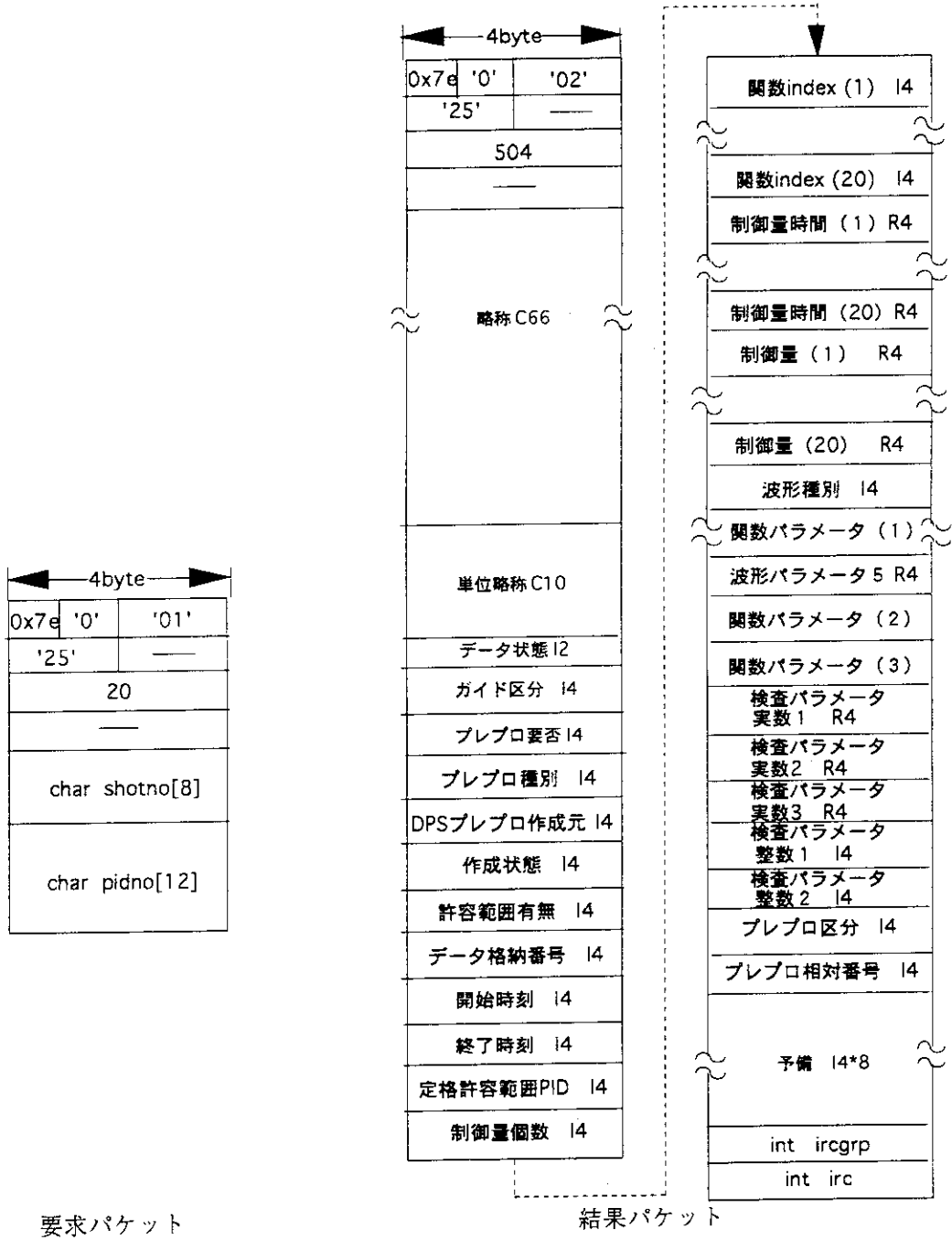
付録6.3図 pontdt パケットフォーマット



付録6.4図 tserdtパケットフォーマット



付録6.5図 shcdtパケットフォーマット



付録6.6図 prprdtパケットフォーマット

## 付録7 資格検査テーブルの構造

資格検査テーブルは、運用管理者がアスキーファイルを設定し、バイナリーファイルに変換し使用する。

## (1) アスキーファイル

アスキーファイルに設定する項目は、ホスト名称、IPアドレス、許可レベルを1行毎に設定する。アスキーファイルの設定例を以下に示す。

|      |                    |        |                            |
|------|--------------------|--------|----------------------------|
| *    |                    |        | ← 先頭文字が*の行は、コメント行。         |
| znsv | ip = 157.111.141.6 | permit | ← アクセスを許可する（読込のみ）。         |
| expl | ip = 157.111.130.6 | deny   | ← アクセスを許可しない。              |
| znsv | ip = 157.111.130.4 | super  | ← アクセスを許可する（読書可）。          |
|      | ip = 157.111.140.* | permit | ← アクセスを許可する（読込のみ）。         |
|      |                    |        | IPアドレス部の*は、0-255のすべてに適合する。 |
| *    |                    |        |                            |

## (2) バイナリーファイル

IPアドレス部は、アスキーファイルの'|'で区切られたIPアドレスをそれぞれ整数変換して1byteづつ4桁の正整数領域に格納する。

IPアドレス部が'\*'の場合は、変換値を0として対応するマスク部を255として'\*'以外のアドレスが同一の全てを（1～255）許可または不許可とする。

許可レベルは、アスキーファイルの'permit'と'deny'により、'permit'の場合、

1: 許可（読込のみ）、'deny'の場合、0: 不許可とする。

また、アスキーファイルの許可レベルが'super'の時は、2: 許可（読書可）とする。

(1)で示したファイルを変換し、16進数で表現すると以下ようになる。

| IPアドレス<br>unsigned_int | マスク<br>unsigned_int | 許可レベル<br>int |
|------------------------|---------------------|--------------|
| 9d6f8d06               | 00000000            | 00000001     |
| 9d6f8206               | 00000000            | 00000000     |
| 9d6f8d04               | 00000000            | 00000002     |
| 9d6f8d00               | 000000ff            | 00000001     |

付録8 全系DB運用管理画面

コマンドツール (コンソール) - /bin/csh

id. so. 1: screendump: warning: /usr/lib/libpixrect.so.2.1 has older revision than expected 1.4

[無変換]

時計

1997年04月18日

JST

CPU

100

選択項目

全系DBファイル管理  
 資格検査ファイル設定  
 設定値変更

[無変換]  
メニューを選択してください。

全系DBファイル管理

自動蓄積除外中

全体 使用 可能 可能ファイル数 終了

ファイル運用領域 (MB) 30720 16993 13727 498

DB蓄積領域 (MB) 267264 71006 196256 11791

対象ホスト: Hプロク  
蓄積ログ  
MTプロク

操作モード選択 回避 復元 削除 消去 移動

MT回避最大ファイル数 (+) (-) 500 MTデバイス名称: ndbcnr:/dev/nrst12

| AppNo   | 創成日付              | DB | 有無                | 回避日付   | 回避                | ホム | ム    | アクセス日付 | 場所 | 容量 (MB) |
|---------|-------------------|----|-------------------|--------|-------------------|----|------|--------|----|---------|
| E027911 | 96/10/29 10:51:29 | 有り | 97/01/24 01:17:11 | DE2002 | 97/01/17 13:18:16 | 蓄積 | 13.3 |        |    |         |
| E027912 | 96/10/29 11:26:17 | 有り | 97/01/24 01:16:23 | DE2002 | 97/04/08 14:13:26 | 蓄積 | 13.9 |        |    |         |
| E027913 | 96/10/29 11:54:50 | 有り | 97/01/24 01:19:34 | DE2002 | 97/04/08 14:02:29 | 蓄積 | 11.4 |        |    |         |
| E027914 | 96/10/29 13:47:43 | 有り | 97/01/17 13:18:17 | 運用     | 13.2              |    |      |        |    |         |
| E027915 | 96/10/29 14:18:40 | 有り | 97/04/01 22:01:17 | 運用     | 13.1              |    |      |        |    |         |
| E027916 | 96/10/29 14:47:29 | 有り | 97/04/01 22:06:36 | 運用     | 16.1              |    |      |        |    |         |
| E027917 | 96/10/29 16:01:57 | 有り | 97/01/17 13:18:18 | 運用     | 16.0              |    |      |        |    |         |
| E027918 | 96/10/29 16:20:04 | 有り | 97/04/01 22:13:47 | 運用     | 9.9               |    |      |        |    |         |
| E027919 | 96/10/29 16:52:04 | 有り | 97/01/17 13:18:19 | 運用     | 16.0              |    |      |        |    |         |
| E027920 | 96/10/29 17:13:31 | 有り | 97/02/19 21:15:16 | 運用     | 16.6              |    |      |        |    |         |

再送り 頁前進 頁後退

5 | 頁

