講座 はじめてのデータ収集

3. データを集めよう

中西秀哉，奥村晴彦[1)]
（核融合科学研究所，[1)]三重大学教育学部）

Let's Acquire Data!

NAKANISHI Hideya and OKUMURA Haruhiko[1)]
National Institute for Fusion Science, 322-6 Oroshi-cho, Toki 509-5292, Japan
[1)]Mie University, 1515 Kamihama-cho, Tsu 514-8507, Japan

(Received 25 November 2004)

In fusion experiments, diagnostic control and logging devices are usually connected through the field bus, e.g. GP-IB. Internet technologies are often applied for their remote operation. All equipment and digitizers are driven by pre-programmed sequences, in which clocks and triggers give the essential timing for data acquisition. Data production rate and amount must be checked in comparison with the transfer and store rates. To store binary raw data safely, journaling file systems are preferably used with redundant disks (RAID) or mirroring mechanism, such as "rsync". A proper choice of the data compression method not only reduces the storage size but also improves the I/O throughputs. DBMS is even applicable to quick search or security around the table data.

Keywords:
data acquistion, sequence control, field bus, remote participation, clock, trigger, meta-data, block transfer, batch processing, journaling file-system, RAID, rsync, NAS, data compression, DBMS

3.1 計測器とデジタイザを運転する

第2章までのお話で，計測器から得られるアナログ信号をA－D変換するデジタイザ，それを運転・制御するコンピュータが一式そろいました．ここからは，そうした装置を実際に動作させ，本来の目的である計測データを集める手順へと話を進めていきます．

3.1.1 計測器とデジタイザの結線

計測データが流れる経路は，基本的に，

計測器
⇩ … アナログ信号
デジタイザ（A-D変換）
⇩ … デジタル信号
データ収集コンピュータ

となっています．しかし規模の小さいシステムでは，計測器とデジタイザが一体になったデジタル・ビデオ・カメラの利用や，A-D変換ボードをデータ収集PCの拡張バスに挿す形態も一般的です．

こうした結線で最も注意すべき点が電気的絶縁です．高速＝広帯域なデジタル信号線やコンピュータは，微弱なアナログ計測信号にとっては強力なノイズ源です．GND線の接続や筐体の接触などによるデジタルノイズのアナログ系への回り込みは，細心の注意で避けなければいけません．

信号線の絶縁には，アナログの場合は絶縁アンプ，デジタルの場合は光メディア変換器などが必要になります．LANの標準規格であるEthernet用の光変換器は比較的安価に入手できますが，広帯域のアナログ絶縁アンプ等は大変高価です．導入コスト低減のためにも，電気的絶縁が本当に必要か，どの部位で絶縁をとるべきかをよく吟味しましょう．

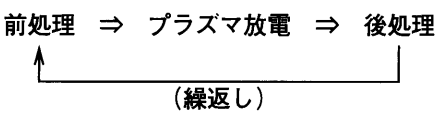
アナログ信号線はT型コネクタにより容易に分岐できますが，末端からの反射が重畳するなど信号波形の崩れる要因にもなります．デジタル信号線も同じなのですが，高周波の矩形信号波形が崩れて'H' = 1が'L' = 0と誤認識されると，伝送データは全く意味を失います．

そのためデジタイザ⇒PC間のデジタル伝送では，信号線そのものを分岐することはほとんどありません．そのかわり同一線路上に複数の装置をつなげられる**バス接続**[1]を用います．計測用バスの詳細は改めて後節でも触れますが，GP-IB[2]やSCSI[3,4]，USB[5]，Ethernet[6]など，分岐方法はバス規格毎に規定されています．よく確認して接続してください．規格外の接続をすると，バスの電気的条件を逸脱してしまい，伝送トラブルに見舞われる危険性が高くなります．

3.1.2 自動制御と実験シーケンス

核融合実験ではほとんどの場合，電源準備等から始まる一連の前準備，実際のプラズマ放電，後処理，を一つの単位として，連続的に多数回繰り返す運転をします．

authors' e-mail: nakanisi@nifs.ac.jp, okumura@edu.mie-u.ac.jp



このようにあらかじめ決められた順序で一連の処理を進行させる自動制御法を**シーケンス制御**と呼びます．様々な前後処理のタイミングを与える（**実験**）**シーケンス**は，通称**シーケンサ**と呼ばれる装置で生成・管理されます．核融合実験は典型的なシーケンス制御ですので，計測器を含めた全装置はシーケンサが出すタイミングで動作します．

古くはシーケンサにはリレー制御盤が用いられていましたが，現在では**プログラマブル・コントローラ（PC）**[*1]とよばれるソリッドステート製品が使われます．いずれにせよ信号の授受は，リレー回路と同じ接点のON/OFF入出力が基本です．接点ON/OFFの時間精度は～数十ms程度と悪いため，次の**3.2**節で述べるように，トリガなど高精度なタイミング信号としては使えません．

大中規模の装置ではシーケンス制御以外にも，プラズマの密度・温度・位置などの自動制御に**フィードバック制御**が用いられます．計測器の信号によって**アクチュエータ（actuator）**への制御出力を変化させ，プラズマの安定保持を図るなどで使われます．

本講座の対象から外れるので，これ以上の詳説はここでは避けますが，計測⇒制御の信号処理にはニューラルネットを応用したものなど，様々な発展形が研究されています．テキストも多数出版されているので興味があれば参照してみてください．

3.1.3 計測器の制御

最近の計測器は多くがインテリジェント化されており，ホストコンピュータとの間でコマンドを授受しながら色々な制御を行うことができます．こうした通信で用いられる伝送路を一般に**フィールドバス**と呼びます．オープンなフィールドバス規格の代表がGP-IBであり，自動車の車載計装用でよく使われるCANバスです．その他，**シリアルバス規格**であるUSB，IEEE1394[*2][7]，Ethernet等も普及しており，100 Mbpsを超える高速通信も可能になっています．

その一方で，RS-232-Cに代表される従来からのシリアル通信も，高速通信が必要ない装置で変らず使われています．いずれにしても計測器との通信路は標準規格であることが多く，WindowsやLinux等OSの標準ドライバから利用可能です．インターネット検索で容易にサンプル・プログラムも入手できるでしょう．また，GP-IBカード等には必ずドライバ・ライブラリが添付されており，サンプル・コードも含まれています．

先ずは簡単なコマンドを送るプログラムを作って試してみましょう．実験シーケンスに則った複雑な制御も，1コマンドを送って返り値をもらう処理の繰り返しに過ぎません．

3.1.4 監視とロギング

計測器やデジタイザは，運転中の機器その他の状態を24時間監視し続ける**運転監視系**と，観測事象が存続している間だけ稼動する**物理計測系**とに大別できます．前者には真空計，後者にはコイル電源，プラズマ波形などが相当します．両者は動作タイミングが全く異なるので，データ保管・取出しを除き，別々の系統に分けたほうが，プログラム開発や障害切り分け等が容易になります．

運転監視系でモニターしている観測値や，機器に加えられた操作を履歴として逐次保存することを**ロギング**と呼びます．真空計のロギングは，長らくペンレコーダで連票紙出力という簡便なアナログ集録でした．しかし小規模な実験でもデータをデジタル化してPCに取り込むのが一般化している今日，ロギングデータだけがオンラインで参照できないのはやはり不便です．計測器のコントローラ類は，フィールドバス経由の制御・データ集録が可能になっているか，あるいは大抵そのオプションがあります．やはり早期にコンピュータ集録へ移行し，他の物理計測データ等と一緒に保管するのが，将来的にも，また共同研究等の利用でも望ましいでしょう．

運転モニタのデータは～1 Hz内外の粗いサンプリングですし，機器の操作履歴はさらに散発的です．このためロギングには，**各サンプルごとに正確な時刻情報を伴っている**必要があります．また一般に，物理計測系が動作する実験シーケンスは標準時刻とは独立しているので，**実験タイミングと標準時刻との対応情報**も忘れずに収集・保存しておかなければいけません．

3.1.5 遠隔操作

Ethernetに代表されるLANが社会全体に普及した現在，

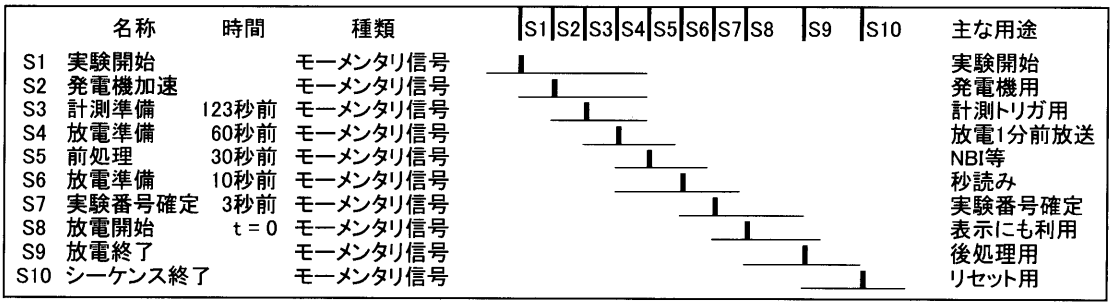


Fig. 1 LHDにおける実験シーケンスの例．主制御信号と呼ばれるS1～S10が，実験の進行タイミングを規定します．

＊1 別称,プログラマブル・ロジック・コントローラ(PLC)です．

＊2 FireWire, i.LINK, DV端子とも呼ばれます．

実験の各種装置もネットワーク経由で遠隔操作するのが一般的になっています．**3.1.1**節の電気的絶縁に関連して触れたとおり，光 Ethernet 規格*3や GP-IB，RS-232-C の光トランシーバを用いれば，光ファイバで距離を大幅に延長できます．利用する光モジュールの種類にもよりますが，300 m 程度の製品が多く，大抵の実験環境には十分な長さです．

LAN (Ethernet) で延長するか，GP-IB や RS-232-C で延長するかは，制御用コンピュータの設置場所やシステム構成にも影響を与えます．前者の場合は制御用 PC を近くにおいて接続し，それをネットワーク経由で別の PC から遠隔操作することになります．これには Windows の**ターミナルサービス（リモートデスクトップ）**やフリーソフトの VNC[8]といった，コンソール画面を遠方に飛ばすツールが便利です．

機器との接続に，専用 PCI カードとケーブルが必須で，光ファイバが使えない場合もあります．付属の専用ユーティリティでないと動かせない機器もあるでしょう．そういう場合にも LAN+リモートコンソールが役立ちます．

便利な反面，**多対多（P 2 P; Peer-to-Peer）接続**が基本である LAN では，他の通信による影響を皆無にはできません．外乱による思わぬ通信遅れや，セキュリティ問題を常に念頭においておく必要があります．フィールドバスを延長する方法では，1 台あるいは少数の接続機器で閉じた回線を専有するので，この心配はあまりありません．

能動機器では誤操作による事故や破壊・破損などの危険性が常にあり，遠隔操作にも高い信頼性が求められます．Ethernet でもスイッチングハブを用いて通信帯域の専有性を上げたり，クロスケーブルで対向接続する手もあります．要は利便性と信頼性のバランスが重要です．

インターネット経由でサイト外の遠隔地から操作をする場合，通信品質は LAN から較べて格段に劣るため，通信エラーを前提としたシステム構成が必須です．**クライアント／サーバ（C/S）モデル**[9]では，ハードウェア制御などの機能を提供するサーバと，依頼を出すクライアントとを明確に分離します．

自律的な保護管理のもとに，サーバ側が全依頼の健全性・正当性をチェックすることで，無許可のアクセスや不正操作を拒否できます．依頼内容が不完全であれば破棄するなど，通信エラーや不意の接続断にも強い仕組みが実現できます．

インターネットで実質的標準となっている TCP/IP[10]では，遠距離通信に付き物である通信エラーの検知や再送などのエラー回復処理が自動化されています．セキュリティにせよ通信エラーにせよ，こうした**ミドルウェア**とよばれる API をうまく選定し活用することで，プログラム開発の負荷が低減されます．

3.1.6 デジタイザの制御と運転

3.1.4節で運転監視系から得られるデータのロギングについて触れたので，ここからは狭義のデータ収集，即ち物理計測系のデジタイザ運転について考えていきましょう．

デジタイザには各種設定パラメータが多数存在しています．多チャネルのものではチャネルごとに設定できる場合が多く，また，新しい製品は一般的に多機能で設定値も多くなります．デジタイザを動作させる前には，それらをすべて運転したい値に設定しなければならないので，前処理はそれだけ長手順になり時間もかかります．

高サンプリング率のデジタイザでは，通常，集録開始のトリガを受ける前から A-D 変換動作を開始しています．トリガ直後の第 1 サンプルを取り落とすことのないようにです．つまり高速デジタイザの前処理には，以下の動作タイミングがあることになります．

1．デジタイザの内部メモリ，レジスタ等を消去（初期化）して運転パラメータを設定する．
2．トリガ待ち状態にする．この時点で A-D 変換は開始される．
3．トリガを受けて，それ以降の，あるいは前後の A-D 変換結果を集録する．

2．以降，A-D 変換結果は内部バッファメモリに書き込まれますが，いわゆる上書きモードで延々と書き換えられています．3．のトリガを受けると，それ以降の上書きを止めて既定時間経過後に A-D 変換を終了します．

こうしたトリガの利用は**STOP トリガ動作**（Fig. 2 参照）と呼ばれます．STOP トリガではトリガ入力直前の A-D 変換結果も保存できるので，遷移現象が起こった前後の状態を記録することが可能です．遷移現象の信号自体をトリガとして A-D 変換を駆動する動作を，特に**イベントトリガ動作**といいます．定常化実験で重要となる技術の一つです．

通常，デジタイザの設定には既定値（default）があるので，実験前に毎回全ての設定値を与える必要はなく，既定値以外の設定のみで十分動作します．しかし次節でも述べるとおり，デジタイザの動作パラメータは，サンプリング

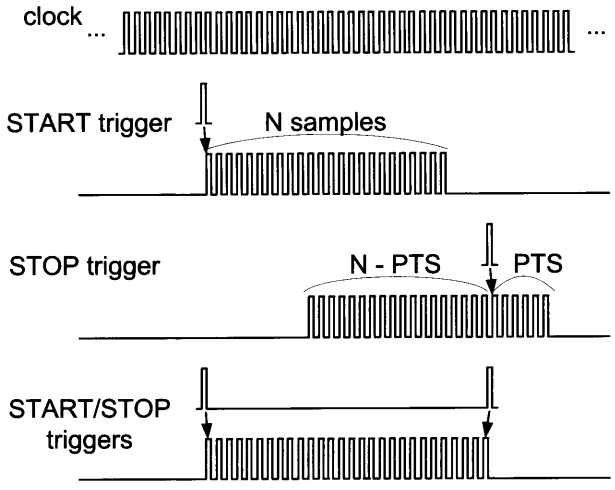


Fig. 2 START，STOP トリガ動作の違い．N はバッファサイズ，PTS は Post Trigger Samples の略．N=PTS にすると START トリガ動作と STOP トリガ動作は同じ結果になります．START/STOP トリガ動作では予めサンプル数は決まりません．

＊3 10/100BASE-FL や 1000BASE-SX/-LX 等が規定されています．

速度やフルスケール，サンプル数など，計測信号を復元するための重要データです．A-D 変換結果の配列データを補完するのですから，実験毎に必ずセットで保存しなければなりません．デジタイザの設定が正しく行われているかの確認のためにも，上記1．の手順でプログラムから全設定値を明示的に設定し，デジタイザから再度設定を読み取って比較検証するのが望ましいでしょう．

デジタイザの後処理動作については次節で取り扱います．

3.2 データを生成し転送する

3.2.1 データとは何か．どう取り扱うのか

実験によって得られるデータといえば，計測器から出力される信号波形や画像などの生データがまず頭に浮かびます．しかしデータとして忘れてはならないのが，本体実験装置や計測器などを運転したときの各種設定パラメータです．

デジタイザが一括して生成する生データとは異なり，機器の運転パラメータはあちこちで分散して与えられます．また，その形態もしばしば多岐にわたるため，運転情報の収集システムを後から開発するのはとても大変です．増設・増強が随時可能な計測器やデジタイザとは異なり，**運転パラメータの収集・保存方法は，システム構築の最初からよく検討しておく**必要があります．

核融合の実験では，通常，実験回数が数万～数十万回と多くなります．どういった実験が過去に行われたのかを把握するためには，機器の運転パラメータが系統的に保存され，検索・閲覧がスムーズにできるようになっている必要があります．こうした目的には，データベース管理システム（DBMS）の活用が好適です．これについては**3.3**節にて解説することにします．

実験データの主役的存在である計測データも，デジタイザが生成する1次元あるいは2次元の整数配列だけでは用をなしません．密度や温度といった物理量に変換するためには，当然，各種の変換係数が必要です．デジタイザの運転パラメータの他に，プリアンプのゲインやセンサーの較正値などがこれに含まれます．これらは生データの**属性**，あるいはデータに対する**メタデータ**と呼ばれ，生の配列データと同じぐらい重要です．

3.2.2 トリガとクロック

真空計など24時間連続で各種状態を監視する常時モニター装置を除き，計測器やデジタイザを運転するには，それらを動作させるタイミングを外部から与えてやる必要があります．最も重要といえるのがデジタイザに A-D 変換開始を指示する**トリガ（trigger）**です．

核融合プラズマの挙動には一般に固有の特性周波数（あるいは周期）があることがよく知られており，電磁流体力学（MHD）的な揺動は数 kHz～数百 kHz といわれます．こうした挙動を計測するには，当然それ以上の時間精度が必要です．つまり一般的な核融合計測の場合，データ集録開始トリガ以降の経過時間は 1 MHz（1 μs）以上の分解能で測られる必要があります．この周期的な時間間隔を精度よく与えるのが**クロック（clock）**と呼ばれるデジタル信号です．一般に ADC ではクロックに同期してデータがサンプリングされますので，デジタルデータの出力は1クロックにつき1サンプルとなります．

数 kHz を超える高速 ADC モジュールの多くは，内部に周波数可変のサンプリング・クロックを内蔵しており，外部／内部クロックを切り替えて利用できるようになっています．内部クロックを用いるとクロック配線の手間が省けて便利ですが，複数モジュールを使って多チャネルの信号を同時計測する場合，同じクロック設定をしても，モジュール間でクロック位相がずれます．周波数が低くなるほど位相差による時間のずれは大きくなりますから，全モジュール共通に単一の外部クロックを与えてサンプリングを同期させる等が必要になります．

他方，コンピュータ内蔵のいわゆるシステムクロックには，通常 100 Hz～1 kHz が使われており，プラズマ計測で用いるサンプリング・クロックよりかなり低い周波数です．このため PC のシステム時計は，A-D 変換開始等の時間精度が必要なタイミング制御には使えません．ご注意ください．

3.2.3 データ転送の開始タイミング

計測対象となるプラズマの持続時間は，核融合実験の歴史の中で長らく1秒未満でした．今でも1分を超える保持が可能なのは，超伝導コイルを用いている TRIAM[11]，LHD[12]等，一部装置の定常化実験に限られています．小型装置では 1 ms 以下の場合もあります．計測用 ADC としては，電圧信号波形を時系列デジタル・データに逐次変換する**トランジェント・レコーダ型**がよく使われます．観測したいプラズマ挙動の周波数帯，例えば流体力学的揺動などにあわせてサンプリング速度は kS/s～MS/s で，kS～MS　程度のデータ・バッファを内部に備えているタイプです．

工場プラントなどの計測制御用途よりかなり高いデータ生成率ですから，データ転送では，通信帯域幅を最大限利用できる方式が求められます．データをある程度の大きさにまとめて一気に送る**ブロック転送**がそれです．逆に，生成されたデータを1サンプルずつ即座に送る方式をとれば，データ到着のリアルタイム性は向上します．しかし通信に占めるオーバーヘッドの割合も大きくなるので，単位時間に送れるデータ量は大幅に下がってしまいます．そのため，実験時間の短かった核融合実験では，実験終了後にデータ転送および解析・表示を一括して行う**バッチ処理**動作が定着しました．

しかし21世紀に入ってようやく，計測器やデジタイザでも Ethernet や USB に代表される安価で高速なデータ伝送路が利用可能になってきました．これまではリアルタイムな連続処理かバッチ処理かによって処理系統を分離して構築・運転していましたが，実時間性と高速性を両立した新たなデータ収集系の模索も始まっています[13]．

さて，放電終了後，データ転送開始タイミングを知る／知らせるにはいくつか方法があります．すでに A-D 変換が終了してデジタイザ中に蓄えられている観測データの転送

開始ですから，基本的に時間精度は必要ありません．

1．放電終了／データ生成完了の信号を受ける（**ハードウェア割込み**）
2．データが生成されたかどうかを繰り返し確認し続ける（**ポーリング**）
3．一定時間経過後（**タイマ**）
4．ネットワーク経由で放電終了情報を頒布（**メッセージ・パッシング**）

2や3の収集プログラムは比較的簡単に作ることができますが，2ではシステム負荷が不必要に上がる，3では実験タイミングの変更に柔軟に対応できない等の欠点もあります．プログラム開発負荷も考慮して，ふさわしい手段を選択しましょう．

3.2.4 データの生成量

トランジェント・レコーダ型ADCの場合，単位時間当たりのデータ生成量は単純にサンプリング速度(周波数)に比例します．1サンプルあたりの分解能によっても変わりますが，よく用いられる10〜16ビット分解能のADCでは，1サンプルのデータ長は2バイトになります．

例をあげて計算してみましょう．100 kHzサンプリングのADCが生成するデータ量は，毎秒

$$2\ \text{byte/sample} \times 10^5\ \text{sample/s} = 0.2 \times 10^6\ \text{byte/s}$$

と1チャネルあたり0.2 MB/sです．モジュール型のデジタイザ・フロントエンド（DFE）を使って1筐体に50チャネル入れると10 MB/sとなります．CCDなどのカメラ計測ではどうでしょう．画面がVGA（幅640高さ480ピクセル）で分解能がモノクロ（グレースケール）10 bit，プログレッシブ・スキャン（60 fps）*4の場合，

$$2\ \text{byte} \times (640 \times 480) \times 60\ \text{/s} = 36.864 \times 10^6\ \text{byte/s}$$

と，カメラ1台で約35 MB/sになってしまいます．

このようにデータの生成率がわかれば，それに計測時間をかけることで生成量が得られます．上の例にならってデータの生成率と生成量を，各計測器あるいはデジタイザ群ごとに計算してみましょう．システム動作への理解が深まるほか，次の増設，設計見直しへの大きな助けになります．

3.2.5 どれくらいまで転送できるか

デジタイザの種類によっては，データ生成率がPCへのデータ転送速度を超えるような運転も一時的には可能です．しかし，転送できないデータが徐々にバッファ内に溜まっていき，いつかはバッファが溢れてしまいますので，この状態を長く続けることはできません．特に，**リアルタイム収集**を行う場合は1系統あたり厳密に，

データ生成率 ≤ データ転送速度

の条件を満たしている必要があります．データ量は無限に大きくなり得ますので，次節で扱う保存の面でも注意が必要です．

これに対して**バッチ収集**の場合，A-D変換結果は内部バッファに蓄えられ，データ転送開始は変換完了後になります．生成率は転送速度からは独立した条件となりますが，データ量はバッファ・サイズが上限となります．ADCの各チャネルで

データ生成率×変換時間＝データ生成量
≤ バッファ・サイズ

となります．転送時間に余裕があるバッチ収集も，当然，次の実験が始まるまでの3分〜5分の有限の時間内に処理を完了させなければなりません．核融合実験ではこれをよく**ショット間処理**と呼びます．

ショット間では，データ転送から解析・表示まで全ての処理が必要になります．それを考えると，時間的に必ずしも余裕があるわけではなく，特にデータ量が多くなった場合は非常に厳しい時間配分を求められます．計測データは全て，

デジタイザ⇒データ伝送路⇒収集プログラム

と転送されます．伝送路にはいろんな種類があり速度も様々ですが，伝送路の通信帯域を最大限有効利用するためには，収集プログラムの処理がデータ転送の足を引っ張らないようにする必要があります．

では具体的にどれぐらいのデータ転送が可能なのか簡単に計算してみましょう．3分周期の実験を想定すると，ショット間処理で使える時間は余裕を見て120秒程度でしょう．計測器I/Fとして一般的なGP-IBは，最大伝送性能こそ1 MB/sながら実行速度は数十kB/sです[1]．仮に30 kB/sならGP-IB経由でデジタイザからデータ収集できる上限は，

$$30\ \text{kB/s} \times 120\ \text{s} = 3.6\ \text{MB}$$

1つのGP-IB系統で約3.6 MBになります．これは切れ目なく120秒間データを送り続けた場合の数値です．もしアプリケーションが1チャネルごとに，**転送⇒解析⇒表示⇒保存**の処理を順次行っていたりすると，転送の時間帯以外はデータが流れません．収集できるデータ量もそれだけ減ってしまいます．

（中西秀哉）

3.3 データの保存と共有

保存・共有すべきデータとは，一般に生データと呼ばれるバイナリーデータと，検索・統計処理の対象となるパラメータ類です．後者はExcel表，あるいはCSVテキストのファイル形式で，実験番号ごとにレコード（行）を蓄積していくことが多いでしょう．こうしたファイル保有は簡便な反面，コピーを繰り返している間に複数バージョンが出来てしまう問題点があります．本節の最後で触れますが，データベースを利用することでこれは解決できます．

データサイズ的には表データは生データに比べてはるか

＊4 fps: frame per second．プログレッシブ・スキャンはノンインターレース・モードともいわれる．通常のTVで使われるのは偶数奇数の走査線を2回に分けるインターレース・モードでNTSC 30 fps，PAL 25 fps.

に少量です．また全実験にまたがるデータを1ファイル中にまとめることで統計処理などの利用価値も上がります．これに対して生データは個々のサイズが大きく，生成される日時もバラバラですので，どうしても多数のファイル群になってしまいます．これらを安全に管理するには信頼性のあるファイルシステムと，冗長性のある保存・複製装置が重要になります．

3.3.1 ジャーナリングファイルシステム

実験室環境ではUPS（無停電電源装置）を使っていても種々のトラブルや人的ミスで不意の電源断が生じることがあります．2004年10月23日の新潟県中越地震でも，停電が繰り返し起こったため，小千谷市のサーバがUPSのバッテリ切れで自動シャットダウンできずクラッシュしたと報じられています*5．

従来のファイルシステムは電源断で壊れやすく，しかも再起動時にスキャンディスクやfsck（UNIX系OSのファイルシステムのチェック）に時間がかかりました．しかし，今はWindowsならNTFS，Linuxならext3やReiserFS，XFS，JFS，Reiser 4といったジャーナリングファイルシステムがあります．電源断があっても，記録をロールバックし，ファイルシステムの整合性を保ってくれます．Solaris 7以降やMac OS X 10.2.2以降でもジャーナリングを設定できます．

最近のLinuxでよく使われているext3は従来のext2との互換性が売りですが，ジャーナリングなしのext2に比べて書き込み速度が若干遅いようです．高速性が要求される場合は他の選択肢を考慮するといいでしょう．なお，ディスクI/Oはメモリにキャッシュされますので，見かけの速度はメモリの量によって大きく変わります．

3.3.2 RAID

複数のディスクを用いて全体としての信頼性や速度を上げるRAID（Redundant Array of Independent Disks，レイド）と呼ばれるディスクアレイ技術がよく使われます[14]．

RAIDには0～5のレベルがありますが，0，1，5がよく使われます．

RAID 0は，2台のHDDにデータを分割して書き込みます（ストライピング）．速度は2倍に近づきますが，信頼性が増すわけではありません．

RAID 1は，2台のHDDに同じ内容を書き込みます（ミラーリング）．速度はやや落ちますが，信頼性が増します．

RAID 5は，n（≥ 3）台のHDDを使い，データを$n-1$台に分割して書き込み，残りの1台にパリティ情報を書き込みます．どの1台にパリティ情報を書き込むかは常に変化します．ストライピングによって速度が改善されるだけでなく，どれか1台が壊れてもデータが失われません．

また，RAID 0+1などと称して，RAID 0をさらにミラー化することもあります．

RAIDはさらにソフトウェアRAIDとハードウェアRAIDに分類されます．一般に，ハードウェアRAIDのほうが高速で，電源を切らずにディスクを入れ替えるホットスワップができるため便利です．ただ，RAIDコントローラが故障して代品が得られない場合は，データがまったく読み出せなくなる可能性もあります．この点ではソフトウェアRAIDのほうが安心です．速度についても，CPUが速くなれば両者の差は縮まります．

3.3.3 S.M.A.R.T.

ディスクのクラッシュには，異常なシーク音やリトライによる速度低下といった前兆を伴うことがあります．

最近のHDDが備えるS.M.A.R.T.（Self-Monitoring，Analysis and Reporting Technology）というモニタ機能を使えば前兆を捉えることがある程度可能です．温度センサを備えたディスクなら，この機能で温度を調べることができます．

UNIX系OSではsmartmontools*6を使って異状が見つかれば管理者にメールが届くように設定しておきましょう．

3.3.4 rsyncを使った同期

本格的なミラーリングでなくても，実験ごと，あるいはUNIX系OSが標準で備えるcronを使って定時に別ディスクにミラーすることができます．具体的には，rsyncコマンドを使って

```
rsync -auv /usr/local/data /opt/local/
```

とすれば/usr/local/dataディレクトリをそのまま/opt/local/dataにコピーします．すでに同じファイルがあれば，新しいものだけが送られますので，単なるcpコマンドより無駄がありません．

データ量が膨大でなければ，ネットワーク経由で別マシンにミラーしておけば，さらに安心です．

```
rsync -auv /usr/local/data foo:/usr/local/
```

とすればローカルマシンの/usr/local/dataディレクトリをリモートマシンfooの同じ場所にコピーします．逆に

```
rsync -auv foo:/usr/local/data /usr/local/
```

とすればリモートマシンfooから更新部分を取り寄せます．これにより二つのマシンの同期をとることができます．

ネットワーク的に遠いマシンの場合は，オプション-zを付ければ，圧縮して送り，リモートマシンで伸長します．圧縮ライブラリzlibが使われますが，これは後述のように必ずしもバイナリデータの圧縮に適当な方式ではありません．

rsyncは無指定ではリモートマシンでリモートシェルrshを起動しますが，最近はセキュリティ上の理由でこれを標準では許可していないはずです．その場合は，オプション`-e ssh`を追加するか，環境変数`RSYNC_RSH`にsshを設定すれば，安全なシェルssh経由で（暗号化して）送られます．より性能を上げるにはrsyncをサーバとして起動しておきます．詳細は`man rsync`でマニュアルをご覧ください．

cronコマンドを使った定時バックアップについては，

*5 http://itpro.nikkeibp.co.jp/free/NCC/NEWS/20041106/152236/

*6 http://smartmontools.sourceforge.net/

cronやcrontabについて，やはりmanコマンドで調べてください．

WindowsのUNIX互換ツール群Cygwinにもrsyncやsshコマンドが含まれていますが，本物のUNIX互換OSより性能が出ません．Windows同士なら単にファイル共有を使うのが簡単で高速です．Windows・UNIX間なら，UNIX側でSambaを立ち上げてファイル共有するか，あるいはFTPやSCP，WebDAV等を使います．逆に，Windows側の共有フォルダをUNIX側からsmbclientで使うこともできます．

3.3.5 FTP，SCP，WebDAV

FTPは昔から使われてきたファイル転送プロトコルです．ダウンロードには今ではHTTPのほうが便利ですが，アップロードにはまだよく使われます．FTPでは認証パスワードが暗号化されずにネットに流れるので，安全でない場所を通過する場合はssh(scp)を使いましょう．WindowsではWinSCPがよく使われます．

最近はアップロードにもHTTPプロトコルを使うWebDAVが流行しています．シェアNo.1のApache（アパッチ）Webサーバも，2.xでは標準でWebDAVをサポートしています．SSLを併用すればパスワードもデータも暗号化されますが，実験データではSSLは大袈裟かもしれません．やはりApache2.xで標準サポートされたDigest認証を使えば，パスワードだけ安全な形でやりとりされます．

3.3.6 NAS

ファイル共有用に別マシンを立ち上げなくても，今ならNAS（Network Attached Storage，ナス）を使うのが手っ取り早いでしょう．数万円程度でもギガビットLANに対応したものがありますが，TCP/IPやSMBプロトコルのオーバヘッドのため，単純なUSB 2.0接続の外付けディスクの速さにはなかなか及びません．非TCP/IPのNDAS(Network Direct Attached Storage)プロトコルを使ったものは，より高速になります．

より高価なものは，RAID，ホットスワップに対応しています．

3.3.7 データ圧縮

データ圧縮は単にディスク容量を節約するためだけのものではありません．CPUに余力があれば，データ圧縮をすることにより，ディスクやネットワークの見かけのスループットが上がります．要は，CPUとI/Oのバランスの問題です[15]．

例えば，12ビットのADC出力を2バイトとして扱わず，二つで3バイトとするようなトリビアルな圧縮でも，データ量が3/4になり，見かけのI/Oのスループットは33%向上します．

こういったADC出力は，Zipやgzipやzlibライブラリといった汎用ツール[16]で圧縮しても，CPUにかける負荷の割には，あまり縮みません．その理由は，これらのツールが，繰り返し現れるバイト列を見つけるために大部分の時間を割いているからです．ADCの生出力に同じデータ列が繰り返し現れることは，人間の書いた文章ほどは期待できません．また，汎用ツールはバイト単位でデータを見ていきますので，バイト単位でないデータは不利になります．

あまり動きの激しくないチャンネルについては，時間軸に沿って差分を取れば，0に近い値が増えるために，縮みやすくなります．このとき，例えば16ビット値(−32768〜32767)同士を引き算すれば範囲は17ビット（−65535〜65535)になるように誤解されるかもしれませんが，引き算はmod 65536ですればいいので，差分も16ビットに収まります．

こういった方法をさらに洗練したものが，音楽データ(16ビット)のロスレス(lossless，可逆)圧縮ツールとして配布されています．MP3などはロスレスでないので，実験データの圧縮には使えません．

2次元の画像データについては，JPEGはロスレスではありませんので，PNGやJPEG LSなどロスレスの圧縮形式を選びます．JPEG LSはPNGよりずっと軽いロスレス圧縮方式としてもっと使われてしかるべきです．ロスレスでなくていいなら，JPEGよりノイズの目立ちにくいJPEG 2000をお薦めします．

3.3.8 データベースの利用

ADCから取得した大量の生データなどは，通常のバイナリファイルとして保存すればいいでしょう．

実験のパラメータなどのメタデータをどう保存するべきかは，ケースバイケースです．管理データはテキストファイルで保存するのがUNIXの伝統であり，そのほうがgrepやPerlなどで簡単に扱えますし，万一ファイルが壊れても，生きているデータを拾い出せる可能性があります．逆に，不馴れな人にテキストエディタで編集させると必ず壊されてしまうのも事実です．専用のツールでないと編集できないようにして，編集時に入力チェックを十分するように作っておくほうが安全です．

データ入力ツールを作る際は，ファイル入出力を自前で行うより，データベース管理システム(DBMS)を使うほうが楽です．同時に更新をかけてもデータが壊れたりしませんし，大量のデータでもインデックスを生成することにより高速に検索できます．

よく使われるオープンソースのDBMSには，機能が豊富なPostgreSQL[17]，高速なMySQL[18]，簡便なSQLite[19]などがあります．SQLiteはまだ知名度が低いかもしれませんが，PHP 5に同梱されて有名になりました．大規模システムには使えませんが，バックグラウンドでプロセスを動かす必要がなく，単なるファイルを扱う気軽さで使えるので便利です．

（奥村晴彦）

参考文献

[1] H. Nakanishi, M. Kojima *et al.*, J. Plasma Fusion Res. **72**, 1062 (1996).
[2] 岡村廸夫：「標準ディジタル・バス（IEEE-488）とその応用」（CQ 出版社，東京，1981）．
[3] インターフェース編集部（編）：「最新SCSI マニュアル」（CQ 出版社，東京，1989）．

[4] 大島啓孝他：「SCSI 完璧リファレンス」オープンデザイン，No.1（CQ 出版社，東京，1994）．
[5] 桑野雅彦他：「USB 機器＆ドライバ実践開発手法」インターフェース，No.1 in 2001, p.51（CQ 出版社，東京，2001）．
[6] 好川哲人：「LAN 技術総合入門」オープンデザイン，No. 12（CQ 出版社，東京，1996）．
[7] 北山洋幸他：「IEEE1394 のハード＆ソフト入門」インターフェース，No.7 in 1999, p.51（CQ 出版社，東京，1999）．
[8] *VNC* http://oku.edu.mie-u.ac.jp/~okumura/linux/?VNC (2004).
[9] 吉岡隆一，山本哲夫，松尾紀彦，須藤隆成：「C/S ネットワーキング」(日経 BP 出版センター，東京，1995).
[10] D.E. Comer and D.L. Stevens,：「TCP/IP によるネットワーク構築 Vol. III」（共立出版，東京，2003）．
[11] E. Jotaki and S. Itoh, J. Plasma Fusion Res. **73**, 330 (1997).
[12] H. Nakanishi, S. Hidekuma, M. Kojima *et al.*, J. Plasma Fusion Res. **72**, 1362 (1996).
[13] H. Nakanishi, M. Kojima and LABCOM group, Fusion Eng. Des. **56-57**, 1011 (2001).
[14] 日高亜友：「ローダブル・カーネル・モジュールを使った計測データの蓄積テクニック」インターフェース，No. 11 in 2004, p.96（CQ 出版社，東京，2004）．
[15] 奥村晴彦：「データ圧縮の基礎から応用まで」C Magazine, No. 7 in 2002, p.13（ソフトバンク，東京，2002）．
[16] 奥村晴彦，山崎敏：「LHA と ZIP：圧縮アルゴリズム×プログラミング入門」（ソフトバンク，東京，2003）．
[17] PostgreSQL http://www.postgresql.org/
[18] MySQL http://www.mysql.com/
[19] SQLite http://www.sqlite.org/

奥村晴彦（おくむらはるひこ）

物理と計算機を行ったり来たりしていましたが，2004年4月からは家から自転車で通える三重大学の教育学部情報教育課程に移り，なんとなく情報教育が専門になりつつあります．

中西秀哉（なかにしひでや）

核融合科学研究所助手．1995年東京大学大学院原子力工学専攻博士課程単位取得退学．博士(工学)．主な研究分野は，実験情報システム，特にデータ収集・処理，計測制御工学等です．素粒子や天文のように，核融合分野からも新たな計算機技術を生み出せないかと日々模索していますが，当面の目標は SLAC のデータ量を LHD で抜くことです．ITER は LHC を超えられるでしょうか．