計算機アーキテクチャ 139−14
（2000. 8. 4）

NICを活用したネットワークRAID方式の提案

松本 尚

ネットワーク環境で有休資源となっているローカルディスクを有効活用して、低コストで高信頼性分散共有ファイルシステムを実現するネットワークRAID方式を提案する。通常のRAIDは安価なディスク装置を複数台使って冗長性を増して信頼性を上げているのに対して、ネットワークRAIDでは安価な計算機を複数台使って冗長性を増して信頼性を向上させる。計算機自体の障害も考慮に入れているため、障害検出訂正方式にはミラーリング方式を拡張した多数決方式を採用している。多数決方式によるデータ認証オーバヘッドコストは大きなものであるが、NIC (Network Interface Card) 内のチェックサム機能を活用することでこのコストを大幅に軽減できる。

The Network RAID System Exploiting Mechanisms in NIC

TAKASHI MATSUMOTO

We propose a brand-new architecture for low-cost distributed file systems with fault-tolerant abilities. We call it "Network Raid System" which does not require dedicated-hardware for conventional RAID devices. Whereas the conventional RAID consists of multiple inexpensive disk drives, the network RAID consists of multiple inexpensive PCs/WSs that are members of a distributed environment and include some disk drive(s) inside. The network RAID adopts a new error detection method "majority decision" which is an extension of the mirroring method. Although authentication cost of the majority decision method is expensive, it can be relieved by exploiting checksum hardware of the NICs.

1. はじめに

パーソナルコンピュータおよびワークステーションの価格性能比の改善と低価格化は著しく、オフィスや工場や研究所はもちろん家庭やSOHOにも複数台の計算機が設置されるようになってきている。現在の計算機はアプリケーションや操作環境整備のためプログラムおよびデータの格納場所としてハードディスク装置を使用している。複数台の計算機を使用する場合に、分散共有ファイルシステムによってハードディスク装置を共有できると、一台ごとにディスク内容を整備する必要がないためにマシン管理コストが大幅に低減できる。また、計算機間のデータの移動が共有ファイルによって自然に行われるため、テープやフロッピーディスクを介したデータ移動よりも大幅に手間が少ない。また、明示的に通信プロトコルやファイル転送プログラムによってデータ転送するよりも簡単である。分散共有ファイルシステムでは十分に高速なプロトコルを使用して多くのシステムではデータ転送に必要な時間コストも十分に小さく抑えられている。これらの利点により、複数の計算機を導入している環境では、分散共有ファイルシステムが導入されている。クラスタシステムでは各ノードにおけるデータの共有は、システムの使い勝手を向上させるのに必須であるため、分散共有ファイルシステムが必要不可欠である。逆に、疎粒度で分散並列実行可能なアプリケーションであれば、分散共有ファイルシステムのみを通信同期手段としてクラスタを運用することが可能である。

分散共有ファイルシステムは他のマシン上のハードディスクを仮想的に使用可能にする機構であるため、プログラムやデータは最終的にハードディスク装置内に保存される。ハードディスク装置は物理的な高速稼働部分が多いため装置自身がクラッシュしたり、非常に高密度の磁気記憶を行っているため誤り訂正不能なデータ化けが発生する確率が無視できない大きさで存在する。重要なデータや時間をかけて整備した操作環境をハードディスク装置の故障から安全に守ることもシステムとしては非常に重要である。また、ハードディスク装置内に内蔵されている半導体メモリによるキャッシュにエラー訂正機能がなく、このレベルでデータ化けを起こすといったトラブルも存在する。ハードディスク装置上データの安全性向上のために近年普及しつつあるのが、RAID (Redundant Array of Inexpensive Disks) 装置である [1]。RAID装置では安価なハードディスク装置を複数使用して、プログラムやデータを冗長な形で格納しておき、1台（もしくは少数台）のディスク装置が故障しても正しい情報が復元可能なディスク装置である。

† 東京大学 大学院理学系研究科 情報科学専攻
Department of Information Science, University of Tokyo
科学技術振興事業団さきがけ研究21「情報と知」領域
PRESTO, Japan Science and Technology Corporation

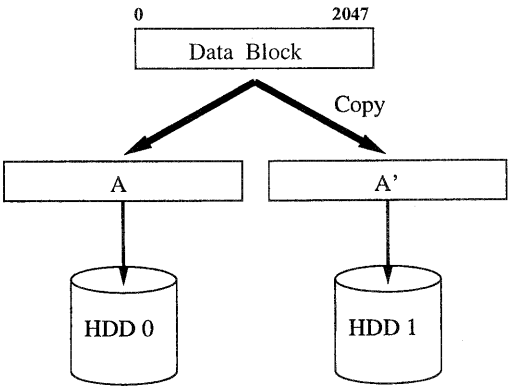

図1　ミラーリング方式

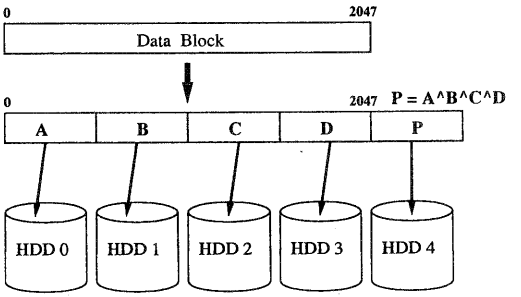

図2　パリティ方式

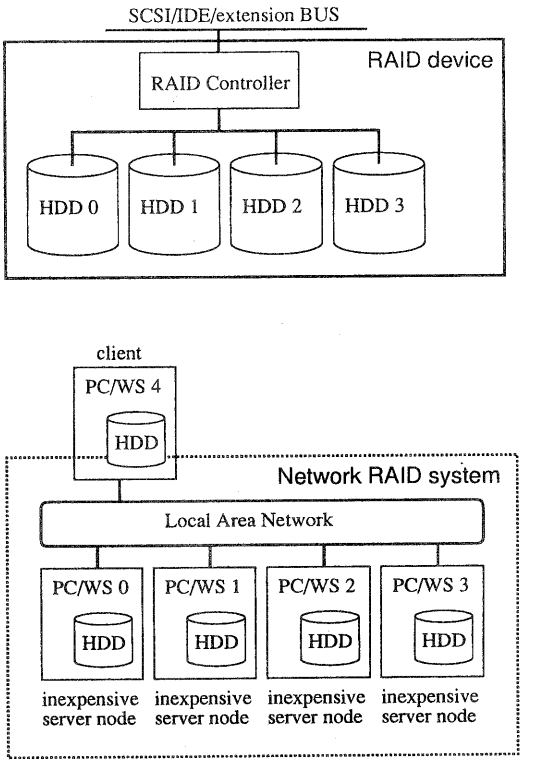

図3　RAID vs. Network RAID

また、複数のディスク装置を同時にストライピングアクセスすることにより、ディスク装置としての性能も単体ディスクよりも向上させることが可能である。

冗長な形への変換方式として、ミラーリング方式とパリティ方式が主に使用されている。ミラーリング方式はデータ処理に負荷をかけないことを重視した方式であり、ディスク装置に格納すべきデータの完全なコピーを他のディスク装置にも格納する（図1）。パリティ方式はディスク容量の有効活用を指向した方式であり、例えば2048byteのデータブロックを512byteずつ4つのサブブロックに分け、さらにサブブロック間の排他論理和による512byteのパリティを計算し、5台のハードディスク装置に分散して格納する方式である（図2）*。元データもしくはパリティを格納したディスク装置が故障しても、故障が1台であれば、他のディスク上のデータから故障したディスクのデータが復元できる。

ミラーリング方式のRAIDはディスクに格納するデータに関して変換をかけていないため、メインCPUのソフトウェアによる実現が低コストで可能であり、ソフトウェアで実現されたシステムも多く存在する。これに対して、パリティ方式のRAIDはパリティ生成を行うオーバヘッドが大きいため、ソフトウェアレベルの実装では性能が低下してしまう。コストが掛かる専用ハードウェアの採用を避けてミラーリング方式によるソフトウェアRAIDを採用したとしても、通常のSCSIやIDEで接続されたディスクに対してシステムを停止させずにディスク交換を行うこと（ホットスワップ）は不可能である。これらの理由から、可用性と信頼性が求められる環境では、RAID装置は専用ハードウェアによって構成され、結果として高価なものとなっている。

本稿では、ネットワークレベルでRAIDの考え方を導入し、分散共有ファイルシステムを低コストで高信頼性化するためのネットワークRAID方式を提案する。

2. 計算機アレイによるRAID

分散共有ファイルシステムは複数の計算機によって共有されるため、ディスク故障等によってシステムがダウンすることの影響が非常に大きい。分散共有ファイルシステムには高い信頼性と高い可用性が不可欠である。このため、分散共有ファイルシステムのディスク装置として専用ハードウェアを搭載したRAID装置を導入して信頼性を向上させる方式が一般的になりつつある。しかし、この場合はRAID装置が接続される計算機（ファイルサーバ）に信頼性と性能が要求され、システムのコストが高くなってしまう。本稿では安価になった普及型の

* parityを格納するディスクをデータブロック毎に変えて分散させる方式もある。

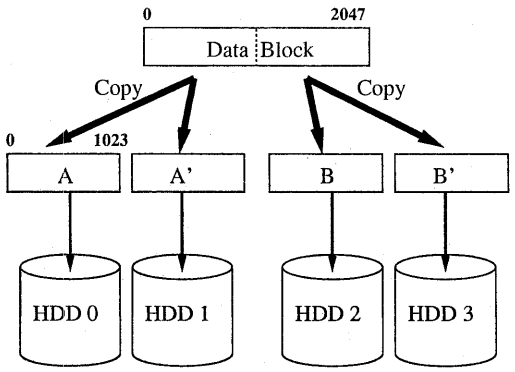

図4　ストライピングを併用したミラーリング方式

パーソナルコンピュータやワークステーションを活用してファイルシステムの信頼性を向上させる方式を提案する。つまり、RAIDは安価なディスク装置を複数台使って冗長性を増して信頼性を上げていたのに対して、ネットワークRAIDでは安価な計算機を複数台使って冗長性を増して信頼性を向上させる（図3）。ファイルシステムの信頼性向上目的のみを目指して新たに計算機を増設したとすれば実現コストが大きくなるが、分散共有ファイルシステムを必要とする環境には元々複数台の計算機が存在する。これらの資源を活用してファイルシステムの信頼性を向上させれば資源の有効活用であり、全体としてのコストは増大しない。計算機レベルで冗長性を持たせることにより、RAID装置の高コストの原因である、電源の多重化、ファンの多重化が自然に達成され、1台の計算機の電源を落しても他の計算機には影響がでないため活線挿抜のための専用回路も不要である。

3. 障害検出訂正方式の検討

ネットワークRAIDに限らずRAID一般についてミラーリング方式とパリティ方式をコスト面から比較した場合、近年のハードディスク装置の大容量化とビット単価の低下によって専用ハードウェアが性能上不可欠なパリティ方式の方が不利である*。

パリティ方式はストライピングアクセスが方式自体に組み込まれているため、単純なミラーリング方式よりも性能面では有利である。しかし、ミラーリング方式もストライピングを併用することで性能向上を図ることが可能である（図4）。実装上の問題を除けば、ミラーリング方式とパリティ方式の本質的な差はディスク容量の利用率のみということになる。複数の計算機からなる分散計算機環境に分散共有ファイルシステムを導入して、プログラムやデータの共有を推し進めると、各計算機のディスク容量は余り気味になることが多い。これらの余剰ディスクを計算機資源と一緒に有効活用するのがネットワークRAID方式であるため、ディスク容量の利用率は大きな問題とはならない。このため、実装が低コストかつ簡単なミラーリング方式もしくはミラーリングを拡張した方式がネットワークRAIDに適している。

ミラーリングをネットワークレベルに拡張した方式として、複数台（2台）の計算機上のディスクに同一のデータを格納しておき、ディスク障害が発見された場合のみ予備のデータを他の計算機から獲得するという方式が考えられる。しかし、安価な計算機を冗長に使用する場合は、ディスク装置の故障のみではなく計算機自体のデータ化けやクラッシュも無視できない。計算機自体のエラーまで考慮に入れて信頼性を考えると、前述のRAIDのミラーリングを単純にネットワークを介して実現した方式では不十分である。計算機レベルでエラーが発生している可能性がある場合は、クライアント側（ファイルをアクセスする側）でデータ読み出し時にデータの正しさを確認する必要がある。つまり、データ読み出し時にミラーされた計算機（サーバ）から複数のコピーを読み出して、本当にデータが一致しているか確認を行う必要がある。データ使用時にデータ不一致が発見されても、ディスク故障以外の場合、どのサーバから送られたデータが間違えているか確定できない可能性があるため、ミラーの数は最低3セット必要であり、「多数決原理」によって正しいデータを決める必要がある**。多数決原理で正しいデータを確定するため、本方式を今後「多数決方式」と呼ぶことにする。クライアント側におけるデータの一致確認を単純にソフトウェアで実現すると、複数のミラーサーバから到着したデータを個別に比較することになり、ミラーリング方式を採用したメリットがなくなってしまう。この問題の解決方法を次節に述べる。

4. ネットワークRAIDの実装方式

前節において、ネットワークRAIDではクライアント側において多数決による格納データの認証を行う必要があることを述べた。クライアント側が複数のミラーサーバからデータを取り寄せて、データを逐一比較していたのではクライアント側のメインCPUに掛ける負荷が大きくなってしまう。また、ネットワーク（LAN）のバンド幅が小さい場合には、複数のミラーサーバからデータを転送することが分散共有ファイルシステムとしてのボトルネックになりかねない。さらに、複数のミラーサーバと通信を行うこと自体が通信オーバヘッドを引き起こし、システム全体の性能を低下させる可能性がある。これらの問題に対する解決策／改善策となる実装方式について本節で述べる。

4.1 データ転送量を削減する方式

データのディスクへの書き込みはすべてのミラーサーバに反映される必要があるため、ネットワーク上のデータ転送量を削減することは原理的に不可能である。しか

* 多くのパーソナルコンピュータにパリティ方式のRAIDが搭載されるようになれば、量産効果によりパリティ方式の専用ハードウェア部分は無視可能なコストになると思われるが、現状はそうなっていない

** ディスクレベルではCRCチェック等によりエラーが発生したディスクのセクタが特定できるが、書き込み前にECCなしメモリやキャッシュ上でデータ化けが発生した場合等は、冗長度が2ではエラー発生側を特定できない。

し、読み出しに関してはデータ転送量削減の可能性がある。データの誤りが高い確率で検出できれることがクライアント側の認証の目的である。そこで、サーバが読み出したデータに対応する認証データ（チェックサムやMD5 [2]やSHA-1 [3]等のハッシュ関数）を一定サイズ（通信パケットサイズ）毎にサーバ側で計算して、1台のサーバはデータと認証データをクライアントに送信し、他のサーバは認証データのみを送信して、クライアントは認証データが一致していることを確認する。そして、認証データが一致していれば、データも一致していると信用してデータを使用する。認証データのbit幅を大きくすればデータ誤りの検出率を高くすることができる。この方式はクライアント側の処理の一部をサーバ側に分散する方式でもあり、サーバにMD5やSHA-1計算用のハードウェアアクセラレータがある場合に特に適している。これらのアクセラレータはセキュリティ対策のために高価なサーバマシンには徐々に普及しつつあるが、ネットワークRAIDがターゲットとするような安価なパーソナルコンピュータには現状では実装されておらず、将来的にそこまで普及するかどうかは定かではない。ただし、ネットワークバンド幅不足がネットワークRAID方式のボトルネックとなっている場合には、この実装方式は非常に効果が高い。

4.2 NICを利用して認証負荷を削減する方式

ネットワークRAIDは分散共有ファイルシステムを対象としているため、本質的に計算機間の通信が発生する。事実上の標準通信プロトコルであるTCP/IPおよびUDP/IPにはデータのチェックサム機能が定義されているため、最近の多くのネットワークインタフェースカード（NIC） [4]にはチェックサム計算支援用ハードウェアが搭載されている。各サーバは生のデータをネットワーク経由でクライアントに転送すると、各サーバのNICはチェックサムを計算して通信パケットに付与する。このチェックサム値（16bit）を認証データとして流用して、クライアント側ではチェックサムが一致していること☆によりデータが一致している見做す。なお、チェックサム計算支援ハードウェアを持たない場合でも、分散共有ファイルシステムの通信プロトコルとしてTCP/IP（またはUDP/IP）を用いる場合は通信レイヤで計算されるチェックサムを流用することが可能である。

MBP2 [5]のような先進的なネットワークインタフェースではMD5やSHA-1といったハッシュ関数を多ビットで計算するハードウェア機能を持っているため、これらを使えばより確実に一致性が確認できる。この実装方式によってデータ認証のためのCPUオーバヘッドをクライアントとサーバの両側においてほとんど無くすことが可能である。

チェックサムが一致してエラー発生が確認されない通常ケースでは、複数のサーバノードからデータ本体を送信することはネットワークバンド幅の浪費である。既存のNICを流用する場合にはこのオーバヘッドはやむを得ないが、ネットワークRAID対応NICを新たに開発する場合は、チェックサムだけ転送できるような通信オプションを用意することで、この無駄を無くすことができる。ただし、Gigabitクラスの高速ネットワークでは、通信オーバヘッドによってバンド幅を使い切ることが難しいため、この程度のバンド幅の浪費は問題とならない。

また、多数決方式といっても不整合が発見されるまでは、2ノードのデータの一致のみで運用を続けられるので、クライアントにデータ（もしくは認証データ）を転送するのは2台のサーバで十分である。2ノードしかデータ（もしくは認証データ）を転送しない場合はアクセスが一部のミラーサーバに偏っていると、データ誤りを長期間見落として放置してしまう可能性が高くなる。このことは回復不能なエラーの発生率を高めるので、アクセスするミラーサーバの選択制御に十分な注意が必要である。

4.3 通信オーバヘッドを削減する方式

ネットワークRAIDではクライアントによるデータ認証を行うため、1台のミラーサーバ以外からは認証用データしか送られてこずに通信量が抑制されたとしても、高信頼化しない場合に対して通信回数はデータ書き込みで3倍、データ読み出しで少なくとも2倍に増えてしまう。そのために、通信自体のオーバヘッドを削減することは性能向上のための重要要素である。筆者が1996年に考案したMBCFプロトコル [6] [7]を通信プロトコルに使用すれば、カーネル内データコピーやカーネルからネットワークRAID実現プロセスへのデータコピーを撤廃することができるためオーバヘッドを大幅に抑制できる。また、通信パケットの到着保証と順序保証を行っているため、分散共有ファイルシステムの層において通信パケットの再送制御等の管理が不要であり、プログラム開発が容易になる。

5. ネットワークRAIDファイルシステム

5.1 NRFSの概要

ネットワークRAID方式をベースに高信頼分散共有ファイルシステムであるネットワークRAIDファイルシステム（**NRFS**）を作成することが最終目的である。ネットワークRAID方式はその上に構築される分散共有ファイルシステムの方式に特別な制限を設けることはない。そこで、既存ファイルシステムとの互換性を重視して、分散共有ファイルシステムの事実上の標準であるSun Microsystems社が開発したNFS (Network File System) [8]をベースにしてNRFSを開発する。

NFS/UNIXでは他の計算機のファイルシステム階層構造の一部をマウント（mount）操作により自分のファイルシステム階層構造の一部として使用可能にする。例えば、nfs_clientというマシンにおいて

```
mount nfs_serv:/usr/local /usr/local
```

というコマンドをroot権限において実行すると、nfs_clientの/usr/localディレクトリ以下にnfs_serverの/usr/local以下の階層構造を仮想的に接続することが

☆ TCP/IPやUDP/IPで通信を行った場合はチェックサムの範囲がヘッダの一部を含むため、若干補正計算を行う必要がある。

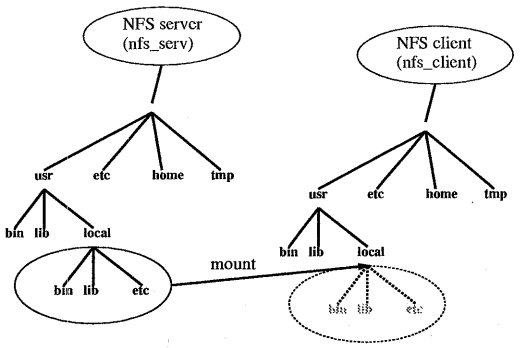

図5 NFSのマウント操作

可能になる（図5）。

```
mount nfs_serv:/usr/local /usr/mylocal
```

のように異なったディレクトリ名の下に接続することも可能であるが、nfs_servの/usr/local以下にあるファイルへのアクセスパスがnfs_servとnfs_clientで異なったものとなるため、分散共有ファイルシステムとしては好ましくない。リモート計算機のファイルシステム階層構造の一部をマウントするディレクトリ（nfs_clientの/usr/localや/usr/mylocal）のことをマウントポイントと呼ぶ。NFSにおいて一つのマウントポイントに対しては一つのリモート計算機のファイルシステム階層構造の一部がマウント可能であり、すでにマウントされているマウントポイントに対してマウント操作を行うと、エラーとなってマウントコマンドは実行されない。

NRFSはマウントポイントにおける制限を緩和することによって、複数のミラーサーバを一つのマウントポイントに対して設定可能にする。例えば、nrfs_clientというマシンにおいて

```
mount nrfs_serv1:/usr/local /usr/local
mount nrfs_serv2:/usr/local /usr/local
mount nrfs_serv3:/usr/local /usr/local
```

の三つのマウントコマンドを受け付ける（図6）。ただし、マウントポイントにすでにマウントされているファイルシステム階層構造がある場合には、新たにマウントされる階層構造と構造が一致していること（ディレクトリ構成、ファイル名、ファイルサイズ、ファイル属性）を確認する。一致していなければ、どちらかの内容にコピーによって強制的に一致させるか、マウントをあきらめるかオペレータに選択させる。多重にマウントされたファイルシステムはオペレーティングシステムが自動的にNRFSとして取り扱う。NRFSにおけるストライピング動作はマウントコマンドの新オプションとして指定可能にする。

典型的なノーエラー時のファイル読み出しと書き込みは以下のように実行される。

nrfs_client内のプロセスが/usr/local以下のファイルに対して読み出し要求（システムコール）を発行すると、三つのミラーサーバのうちの二つのミラーサーバにファイルデータの読み出し要求がMBCFプロトコルで

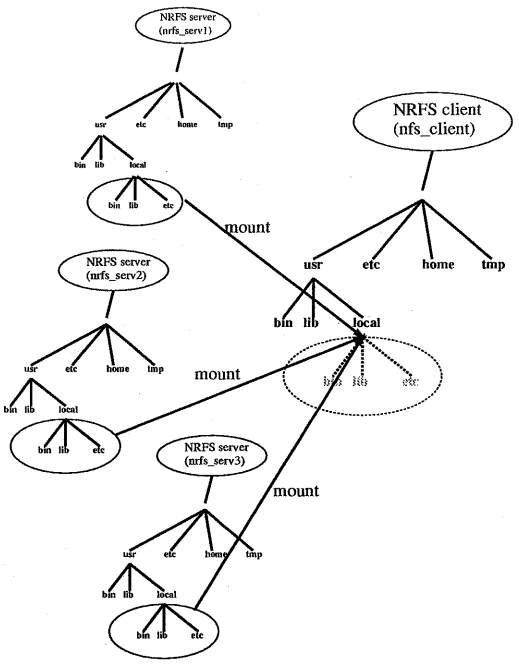

図6 NRFSの拡張マウント操作

送信され、各サーバからデータが認証データ（チェックサム）と共にMBCFで返送される。nrfs_client側においてチェックサムの一致が確認され、読み出し要求を発行したプロセスにデータが渡される。

nrfs_client内のプロセスが/usr/local以下のファイルに対して書き込み要求（システムコール）を発行すると、データ処理単位に端数があってその部分がローカルにキャッシュされていない場合はまずリモートの読み出しを行う。データ処理単位を満たす書き込みデータが揃ったら、三つのミラーサーバすべてにファイルデータの書き込み要求が書き込みデータとともにMBCFプロトコルで送信され、各ミラーサーバではファイルキャッシュもしくはローカルディスクに書き戻される。

5.2 NRFSの構成

NRFSはNFSの拡張として実装されるため、VFS（Virtual File System）がサポートするファイルシステムの一種類としてオペレーティングシステムに組み込まれる。図7にNRFSを含むファイルシステムの構成を示す。図では便宜上NRFSクライアントとNRFSサーバが別のノードとして記述されているが、通常は各ノードはファイルシステムのある部分に関してはサーバとして振舞い、他の部分ではクライアントとして振舞う。この辺りの事情はNFSと同じである。

NRFSシステムはクライアント側がNRFS client daemonとMBCF driverで構成されており、サーバ側がNRFS server daemonとMBCF driverで構成されている。

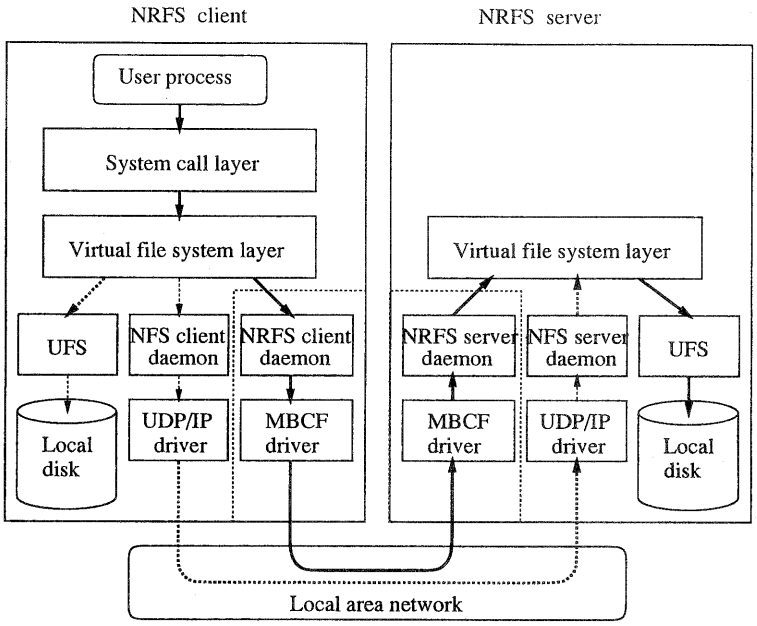

図7　ファイルシステムの構成

6. おわりに

ネットワーク環境で有休資源となっているローカルディスクを有効活用して、低コストで高信頼性分散共有ファイルシステムを実現するネットワーク RAID 方式を提案した。通常の RAID は安価なディスク装置を複数台使って冗長性を増して信頼性を上げているのに対して、ネットワーク RAID では安価な計算機を複数台使って冗長性を増して信頼性を向上させる。計算機自体の障害も考慮に入れているため、障害検出訂正方式にはミラーリング方式を拡張した多数決方式を採用している。多数決方式によるデータ認証オーバヘッドを NIC のチェックサム機能を活用することで大幅に軽減している。ネットワーク RAID 方式に基づくファイルシステムである NRFS を SSS-CORE [9] [10] および Linux に実装する予定である。

謝　辞

ネットワーク RAID に関して御議論いただいた三菱マテリアル株式会社システム事業センター知識産業部の方々に感謝いたします。

参考文献

1) D. A. Patterson, G. A. Gibson and R. H. Katz: A Case for Redundant Arrays of Inexpensive Disks (RAID). In *Proc. of Int. Conf. on Management of Data*, pp. 109–116 (September 1988).
2) R. Rivest: The MD5 Message-Digest Algorithm. RFC 1321, (April 1992).
3) National Institute of Standards and Technology: FIPS PUB 180-1:Secure Hash Standard. U.S. Department of Commerce, (April 1995).
4) Sun Microsystems, Inc.: Fast Ethernet, Parallel Port, SCSI (FEPS) User's Manual Revision 1.0. Sun Microsystems, Inc., (April 1996).
5) 松本 尚, 他: 中粒度メモリベース通信を支援する Memory-Based Processor II, 計算機アーキテクチャ研究会報告, 情報処理学会. Vol.98 No.70 ARC–130–18, pp.103–108 (August 1998).
6) 松本 尚, 平木 敬: 汎用超並列オペレーティングシステム SSS–CORE のユーザレベル通信同期機構. コンピュータソフトウェア, Vol.15, No.3, pp.59–63 (May 1998).
7) T. Matsumoto and K. Hiraki. MBCF:A Protected and Virtualized High-Speed User-Level Memory-Based Communication Facility. In *Proc. of the 1998 ICS*, (July 1998).
8) Sun Microsystems, Inc.: NFS: Network File System Protocol Specification. RFC 1094, (March 1989).
9) 松本 尚, 平木 敬: 汎用並列オペレーティングシステム SSS–CORE の資源管理方式. 日本ソフトウェア科学会第 11 回大会論文集, pp.13–16 (October 1994).
10) 松本 尚, 他: 汎用超並列オペレーティングシステムカーネル *SSS–CORE*. 第 17 回技術発表会論文集, 情報処理振興事業協会, pp.175–188 (October 1998).