

クラスタ型 NAS システム向きディスク使用量制限機能の提案と評価

保田 淑子[†] 沖津 潤[†] 川本 真一[†]
江端 淳[†] 樋口 達雄[†]

クラスタ型 NAS システムの使い勝手と管理容易性を向上させるため、ユーザごとのディスク使用量を制限する機能 XQUOTA を提案した。XQUOTA は、ユーザごとのディスク使用量情報の管理をクラスタ型 NAS システムを構成する各要素 NAS のローカルファイルシステムの持つ Quota 機能に任せ、必要に応じて各要素 NAS からディスク使用量情報を取得することで、クラスタ型 NAS システム全体としての Quota 機能を実現する。各要素 NAS が管理するユーザごとのディスク使用量情報の取得により発生するオーバーヘッドを削減するため、新たに見積り値を管理する方式を提案した。本方式では、ユーザごとのディスク使用量が割当量よりも明らかに少ない場合には、各要素 NAS に問い合わせることなく見積り値を用いて Quota 処理を行う。これにより、複数の要素 NAS に分散するユーザごとのディスク使用量情報を取得するためのオーバーヘッドを削減できる。XQUOTA の有効性を検証するため、我々が提案している仮想一元化 NAS システム X-NAS のプロトタイプに XQUOTA を実装し評価した。その結果、XQUOTA 搭載によるオーバーヘッドを 3% に以下に抑えられ、クラスタ型 NAS システムに適したディスク使用量制限機能を実現できることを確認した。

Disk Quota Method Designed for a Clustered NAS System (X-NAS)

YOSHIKO YASUDA,[†] JUN OKITSU,[†] SHINICHI KAWAMOTO,[†]
ATSUSHI EBATA[†] and TATSUO HIGUCHI[†]

The new disk quota method, what is called XQUOTA, designed for a clustered NAS system has been developed. XQUOTA limits users' disk space of the clustered NAS system by collecting the users' disk space information for each NAS element via an IP network. The users' disk use on each NAS is managed by the disk quota module of the local file system. To reduce the overhead for getting the users' disk use via the IP network, XQUOTA uses the estimate use when the users' disk use is much smaller than the limit. This enables the network overhead to be reduced. To validate XQUOTA, the performance of X-NAS prototype with XQUOTA based on NFSv3 was evaluated. The evaluation results show that the overhead of an eight-way X-NAS with XQUOTA is less than 3% compared to the X-NAS without XQUOTA.

1. はじめに

NAS (Network-Attached Storage) は、急増する電子ファイルを効率良く管理するための IP ネットワークに接続可能なストレージ装置である。様々な OS の稼動するクライアント間でファイルを共有できることから、ファイルサーバやバックアップ用途として市場で広く使われてきている。一般に NAS は複数のユーザで共有される。そのため、一部のユーザが NAS のディスク容量を多量に使用してしまうと、他ユーザが使用できなくなる問題が発生する。また、NAS の大容量化により、管理者がより効率良く NAS を管理し

たいというニーズが高まってきている。このような背景の下、NAS に Quota 機能をサポートすることが一般的になってきている。Quota 機能とは、ユーザごとに使用可能なディスク容量を制限する機能である。この機能により、複数ユーザが NAS を共有する場合でも、各ユーザが使用可能なディスク容量を保証できる。

NAS はディスク容量と価格により、ハイエンド、ミッドレンジ、エントリという 3 つの市場セグメントに分類できる。その中でもエントリ NAS は、従来のストレージ装置に比べて安価で管理も容易なことから、IT 専門の管理者のいないスモールオフィスや企業部門に広く受け入れられている。しかしながら、通常、エントリ NAS はディスク容量を拡張することができない。

この問題を解決するため、我々は、エントリ NAS からミッドレンジ NAS 市場を狙った、安価で使い勝手

[†] 株式会社日立製作所中央研究所
Central Research Laboratory, Hitachi Ltd.

が良く、かつ管理容易でディスク容量を簡単に拡張可能な仮想一元化 NAS システム X-NAS (eXpandable NAS) を提案している^{1)~5)}。X-NAS は複数の安価なエントリ NAS をクラスタ化し、それらを仮想一元化ファイルシステムにより一元管理することで、クライアントに対して仮想的に 1 つの NAS に見せることができる。また、新規 NAS を X-NAS に追加することで、X-NAS 全体のディスク容量を簡単に拡張できる。

X-NAS と同様、ディスク容量を簡単に拡張可能なクラスタ型 NAS システムが提案されているが^{6)~9)}、いずれのシステムも従来 NAS が持つ Quota 機能のサポートについては言及していない。そのため、これらのシステムには Quota 機能が搭載されていないと考えられる。この理由としては、クラスタ型 NAS システムでは、ユーザのファイルが複数の要素 NAS に分散配置されるため、それらのファイルからユーザごとのディスク使用量を取得するオーバーヘッドが大きいことが原因であると考えられる。

一方、分散ファイルシステムや分散ストレージシステムの研究では、分散環境における Quota 機能の実現方法が提案されている。AFS^{10)~12)} は、ユーザごとのファイルを配置するディスクを単一の物理ディスクに限定することで Quota 機能を実現する。また、分散ストレージシステムの 1 つである CFS¹³⁾ では、各ディスクにおいて IP アドレスごとにディスク使用量を制限する Quota 機能を提案している。さらに PAST^{14),15)} は、クライアント側でユーザごとのディスク使用量を管理することで分散システムにおける Quota 機能を実現する。

クラスタ型 NAS システムにおいても、NAS と同様の使い勝手を提供し、管理者が大容量の NAS を効率良く管理するためには、Quota 機能をサポートすることが求められる。しかしながら、上述の分散システムが提案するいずれの Quota 機能も限定された使用方法を想定しておりクラスタ型 NAS システムに適用することが難しい。

そこで、本稿では、X-NAS のようなクラスタ型 NAS システムに適したディスク容量制限機能 XQUOTA を提案する。また、XQUOTA の有効性を検証するため、我々が提案する仮想一元化 NAS システム X-NAS プロトタイプに本機能を実装し、性能評価した結果について述べる。

2. XQUOTA の課題と解決方針

2.1 クラスタ型 NAS システムの構成

本章で考える XQUOTA の対象とするクラスタ型

NAS システムは、複数の要素 NAS を IP ネットワークで接続した構成をとる。ユーザのファイルは複数の要素 NAS にファイル単位またはディスクブロック単位で分散配置される。各要素 NAS はそれぞれサービスデーモンを持つ。サービスデーモンはクライアントからのファイル読み出し・書き込み要求を受け、対象ファイルまたはブロックが実際に格納されている要素 NAS に対しファイルの読み出し・書き込みを行い、その結果をクライアントに返す。また、クラスタ型 NAS システムは、これら複数の要素 NAS を一元管理するための一元管理デーモンを持つ。一元管理デーモンは、管理者からの管理要求を一元的に受け付け、クラスタ型 NAS システムの構成を変更する。

2.2 XQUOTA の課題

Quota は、ユーザごとにディスク割当量を設定し、ユーザのディスク使用量が割当量を超えないようにディスク書き込みを制限する機能である。Quota はユーザごとにそのユーザのディスク使用量を管理し、ユーザがディスクに書き込みを行うたびにディスク使用量を更新する。ディスク使用量が割当量を超えた場合には、ディスクへのデータ書き込みは行わず Quota エラーを返す。

XQUOTA はクラスタ型 NAS システムにおいて Quota 機能を実現する。前述のようにクラスタ型 NAS システムでは、ユーザのファイル群は複数の要素 NAS に分散して配置される。このような分散環境においてユーザのディスク使用量をどのように管理するかが、XQUOTA 実現に向けた課題となる。以下ではユーザごとのディスク使用量の管理方法について検討する。

2.3 ディスク使用量の管理方式の検討

ユーザごとにディスク使用量を管理する最も単純な方法は、ディスク書き込みがあるごとに、各要素 NAS に格納されているそのユーザのデータを探し出し、そのデータサイズを合計する方法である。しかし、この方法は各要素 NAS の探索オーバーヘッドが大きく実用的でない。そこで、ユーザごとにディスク使用量を管理し、これを参照する。この機能を備えたプログラムを用量管理デーモンと呼ぶ。

2.3.1 用量管理デーモン配置方式

用量管理デーモンをクラスタ型 NAS システムの要素 NAS に配置する方法として、特定の要素 NAS に配置する集中管理方式と、すべての要素 NAS に配置する分散管理方式が考えられる。

● 集中管理方式

1 つの要素 NAS に用量管理デーモンを搭載し、そこで全ユーザのディスク使用量を管理する方式

である。ユーザごとにディスク使用量を計算するために、ユーザからの書き込みを常時監視し、書き込みによる変化量を算出する。ユーザの書き込みには、新規書き込みと上書きがある。新規書き込みは書いた分だけディスク使用量が増えるが、上書きの場合、書き込み量と使用量の増加分は一致しない。これら書き込み時に発生するディスク使用量の増減量を求めるためには、書き込みの種類を判別する必要がある。一般に、書き込み要求には書き込みデータとオフセット情報が含まれるが、書き込みの種類を判別するためには、これらの情報のほかに書き込み対象ファイルのサイズ情報が必要となる。そのため、使用量管理デモンはクラスタ型 NAS システム上の全ファイルのサイズ情報を管理する。

● 分散管理方式

各要素 NAS に使用量管理デモンを搭載し、それぞれでユーザごとにディスク使用量を管理する方式である。必要に応じて、ネットワーク経由で各要素 NAS からユーザごとのディスク使用量を取得し、それらを集計し、クラスタ型 NAS システム全体のディスク使用量を算出する。この一連の算出処理を使用量取得処理と呼ぶ。

2.3.2 XQUOTA におけるディスク使用量管理方式の検討

以下では、管理データ規模、実装規模、耐故障性、オーバーヘッドの観点で上記集中管理方式と分散管理方式を比較し、XQUOTA におけるディスク使用量管理方式を検討する。

(1) 管理データ規模

クラスタ型 NAS システムにおいて Quota 処理にともなう管理情報 (XQUOTA 管理情報) を特定の要素 NAS に配置してしまうと、その NAS のハードウェアリソースを圧迫し、ボトルネックが発生しやすくなる。XQUOTA の適用対象とするクラスタ型 NAS システムは要素 NAS として 1TB 程度のディスク容量を持つ エントリ NAS を使用する。オフィスドキュメントサイズは平均 100 ~ 300KB であるため²⁾、1 台の要素 NAS には約 4 千万個のファイルが保持されることになる。集中管理方式ではクラスタ型 NAS システム上の全ファイルのファイルハンドル、サイズ情報とそのファイルの所有者 (ユーザ ID) 等を保持する必要がある。1 ファイルあたり最低 40B の管理情報が必要であると仮定すると、要素 NAS 1 台あたり最低

160MB の XQUOTA 管理情報が必要となる。要素 NAS のメモリサイズは 512MB 程度であるため、数台構成のクラスタ型 NAS システムであってもその管理情報を高速なメモリ上に配置することはできない。一方、分散管理方式では、各要素 NAS に、その NAS 上のユーザごとのディスク使用量情報と全要素 NAS の使用量情報を集計した結果を保持すればよい。エントリ NAS のユーザ数は 1000 人規模であることを考慮すると数 MB の情報しか必要としない。

(2) 実装規模

ソフトウェアの品質を向上させるためには、開発コード量を削減する必要がある。開発コード量の削減により、バグが発生しにくくなり、テスト期間を削減することも可能である。集中管理方式では、ユーザの書き込み要求によってファイルサイズとディスク使用量を管理する処理を新規開発する必要がある。かつ構成台数の増加にも対応させる必要がある。一方、分散管理方式では、各要素 NAS にローカルファイルシステムが搭載されている場合には、既存の Quota 機能を使用して、ユーザごとのディスク使用量情報を管理し、取得することも可能であるため、開発コード量を削減しソフトウェアの品質を向上させることができる。

(3) 耐故障性

集中管理方式では、使用量管理デモンを搭載する要素 NAS が故障すると、復旧時にユーザごとに各要素 NAS に格納されているユーザのデータを探し出し、データサイズを合計してユーザごとのディスク使用量を再計算する。4 千万個 × 要素 NAS 数のファイルを探索して XQUOTA 管理情報を再構築するまでは、クラスタ型 NAS システムの運用を再開できないため長時間運用が停止してしまう。一方、分散管理方式では、いずれかの要素 NAS が故障し使用できなくなった場合にも、復旧時には要素 NAS の使用量管理デモンに問い合わせるだけで、その NAS 上のユーザごとのディスク使用量を求められるため、運用停止時間を最小にできる。

(4) オーバヘッド

集中管理方式では、全要素 NAS のディスク使用量を 1カ所で管理するため、他の要素 NAS に問い合わせる必要がなく Quota 処理を行った場合の書き込み処理のオーバーヘッドが小さい。一方、分散管理方式では、使用量取得処理をネッ

表 1 集中管理方式と分散管理方式の比較

Table 1 Comparison between centralized management and distributed management.

	管理データ 規模	実装 規模	耐 故障性	オーバ ヘッド
集中管理方式	x	x	x	
分散管理方式				x

Quota処理{

```

全要素NASからU_IDに
対応するディスク使用量取得;
ディスク使用量(U_ID)=
Σ(各要素NASのディスク使用量);
current=ディスク使用量(U_ID)+変化量;
if(current>割当量(U_ID))
    Quotaエラー返送;
else
    書き込み処理;
    }
    U_ID: ユーザID

```

図 1 Quota 処理アルゴリズム

Fig. 1 Algorithm for basic quota processing.

トワークを介して行うため、集中管理方式に比べて書き込み処理オーバーヘッドが大きくなる。

集中管理方式と分散管理方式の比較を表 1 にまとめる。分散管理方式はオーバーヘッドが弱点ではあるが、この弱点を克服できれば、他の面では集中管理方式に優れる。そこで、XQUOTA では分散管理方式を採用し、オーバーヘッドを削減する方式を検討した。

2.4 分散管理方式におけるオーバーヘッド

2.4.1 分散管理方式を用いた Quota 処理

図 1 に、分散管理方式を採用した場合の、クラスタ型 NAS システムにおける Quota 処理アルゴリズムを示す。ユーザから書き込み要求を受け付けた要素 NAS は、各要素 NAS に問い合わせ、そのユーザのディスク使用量を取得する（問合せベース Quota 処理）。そして、取得したユーザのディスク使用量を集計し、割当量と比較する。ユーザのディスク使用量が割当量に到達している場合、ユーザからの書き込み要求を破棄し、ユーザに Quota エラーを返送する。ディスク使用量が割当量に到達していない場合、ディスクに対して書き込み処理を行う。

2.4.2 オーバヘッド要因

図 2 に、クラスタ型 NAS システムに分散管理方式の XQUOTA を導入した場合の書き込み要求処理の流れとオーバーヘッド要因を示す。図 2 から、要素 NAS 数に比例して使用量取得処理によるネットワークアクセス回数が増加するため、XQUOTA によるオーバーヘッドが非常に大きくなるのが容易に予想できる。そこで、この使用量取得処理にともなうネットワークオーバーヘッドを削減することが分散管理方式の課題と

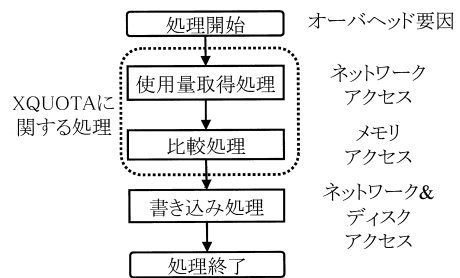


図 2 分散管理方式による書き込み要求処理フローとオーバーヘッド要因

Fig. 2 Flow of write operations and their overheads.

なる。一般に、管理コストの低減とオーバーヘッドはトレードオフの関係にある。そこで、IT 専門の管理者のいないスモールオフィスや企業部門のユーザに対してヒアリングを行い、XQUOTA によるオーバーヘッドが 10%程度であれば、これらのユーザの許容範囲であると判断し、オーバーヘッドを 10%以下に削減する目標を立てた。

2.5 分散管理におけるオーバーヘッド削減方針

一般に、ネットワーク処理による遅延とメモリアクセスによる遅延を比較すると、ネットワーク処理による遅延は数 100 μ 秒オーダーであり、メモリアクセスによる遅延に比べて圧倒的に大きい。したがって、分散管理におけるオーバーヘッドを削減するためには、ネットワークトラフィックを削減することが有効である。分散管理方式におけるネットワークトラフィックを削減するためには、図 2 に示す使用量取得処理の回数を削減すればよい。

Quota 処理において、ユーザごとのディスク使用量が割当量よりも小さいことが明白な場合、ディスク使用量を取得する必要はない。そこで、ディスク使用量の見積り値という概念を導入し、これを使って使用量が割当量よりも明らかに小さい場合を判定し、使用量取得処理の回数を削減する。この方式を見積りベース Quota 方式と呼ぶ。

見積りベース Quota 方式では、新たに見積り値管理デーモンを設ける。見積り値管理デーモンは、ユーザごとにディスク使用量の見積り値を保持する。クラスタ型 NAS システムの各要素 NAS に見積り値管理デーモンを設け、サービスデーモンとペアにして用いる。サービスデーモンはクライアントからの書き込み要求を受けると、見積り値管理デーモンに対して見積り値の更新を要求する。見積り値の更新要求を受けた見積り値管理デーモンは現在の見積り値に定数を加え、その値をサービスデーモンに返す。見積り値の更新に用いる定数は見積り値が実際の使用量を下回らないよ

うに以下の2点を考慮して定める．

- (1) すべての書き込み要求をファイルサイズの変化量が最大となる新規書き込みとして扱う．
- (2) ユーザのファイルはファイルシステムにより管理されるため、ユーザのディスク使用量は実データサイズと管理データサイズの和となる．そこで、見積り値も実データサイズに管理データサイズを加算して算出する．

見積り値を取得したサービスデーモンは見積り値がディスク割当量の $1/n$ (n は見積り値管理デーモン数) 以下であれば、明らかに Quota エラーとならないため、使用量取得処理を実施せず、書き込みを行う．見積り値がディスク使用量の $1/n$ を超えた場合は、使用量取得処理を実施し、実際のディスク使用量と割当量を比較して Quota エラーを判定する．それと同時に、各見積り値管理デーモンに対し、見積り値の修正要求を発行し、見積り値を修正する．

本方式では見積り値がディスク割当量の $1/n$ を超えない限り使用量取得処理を実行しないため、Quota 処理にともなうオーバーヘッドを大幅に削減できる．

3. 見積りベース Quota 方式の実装

次に、見積りベース Quota 方式の有効性を実証するため、この方式を我々が提案する仮想一元化 NAS システム X-NAS に実装する．本章では、まず X-NAS の基本構成と処理方式について述べた後、XQUOTA の実装について説明する．

3.1 X-NAS の基本構成と処理方式

2章で述べたクラスター型 NAS システムは、各要素 NAS がそれぞれサービスデーモンを持ち、それぞれがクライアントからのファイルアクセス要求を処理する．一方、X-NAS は親 NAS と呼ばれる1つの要素 NAS のみがサービスデーモンを持ち、クライアントからのファイルアクセス要求を処理する構成になっており、一般的なクラスター型 NAS システムにおいてサービスデーモンが1つの特殊ケースと考えることができる．

図3に X-NAS の基本構成を示す．X-NAS は、親 NAS と複数の子 NAS から構成される．親 NAS および子 NAS の OS は標準 Linux である．子 NAS は、UNIX 系のファイル共有プロトコル NFS (Network File System) を提供する `nfsd`¹⁶⁾ とデータディスクを含む．各データディスクには、UNIX 系の標準的なファイルシステムを搭載する．親 NAS は、`nfsd` およびデータディスクに加えて、複数 NAS を仮想一元化し、一括して管理するための X-NAS 特有の構成要素 (管

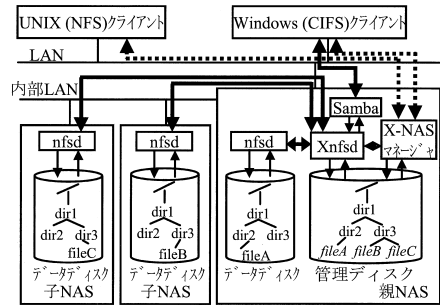


図3 X-NAS の基本構成

Fig. 3 X-NAS overview.

理ディスク, Samba¹⁷⁾, Xnfsd, X-NAS マネージャ) を持つ．

管理ディスクは、クライアントに単一のファイルシステムビューを提供するためにクライアントと同一のファイルディレクトリ構造を持つ．ただし、管理ディスク上のファイルはサイズが0バイトのダミーファイルである．ダミーファイルは、X-NAS が管理する全ファイルに関する属性情報を保持し、ファイルの実体が保持されている要素 NAS のデータディスクを特定するために使用する．Xnfsd は、複数 NAS を仮想一元化するために新規に開発したサービスデーモンである．Xnfsd は、`nfsd` に代わってクライアントからのファイルアクセス要求を受け付け、管理ディスクにアクセスし、その要求のアクセス対象ファイルの実体を保持するデータディスクを特定する．そして、特定したデータディスクを搭載した要素 NAS にファイルアクセス要求を転送し、その要素 NAS 上で稼動する `nfsd` に要求を処理させる．

X-NAS マネージャは、2章で述べた一元管理デーモンに相当し、X-NAS の管理機能を提供する．X-NAS の管理機能には、要素 NAS の追加/削除により X-NAS のディスク容量を増減するオンライン拡張/縮退機能、要素 NAS 間でディスクの残容量を平準化する自律容量リバランス機能がある^{2),4)}．自律容量リバランス機能は、定期的に X-NAS 全体のディスク容量とディスク使用量を監視し、ユーザファイルの追加/削除、あるいはオンライン拡張/縮退により X-NAS 全体のディスク容量バランスが変わった場合に、要素 NAS 間のディスク残容量が等しくなるようにファイルを移動する (自律容量リバランス機能の詳細は文献 2) および 4) を参照していただきたい)．このように X-NAS マネージャは X-NAS 全体としてのディスク容量とディスク使用量を管理する．一方、XQUOTA は、X-NAS 全体のディスク容量管理は X-NAS マネージャに任せ、ユーザごとのディスク使用量のみを管理する．

3.2 XQUOTA のモジュール構成

前述のように X-NAS は 2 章で示したクラスタ型 NAS システムのサブセットであり、親 NAS にのみサービスデーモンが配置されている。そのため、見積り値管理デーモンも親 NAS にのみ配置する。使用量管理デーモンはそれぞれ要素 NAS のディスク使用量を管理する必要があるため、親 NAS および全子 NAS に配置する。

図 4 に XQUOTA を備えた X-NAS の構成を示す。XQUOTA は、割当量管理テーブル、見積り値管理テーブルを含む見積り値管理モジュール、使用量取得モジュールおよび rquotad により実現される。見積り値管理モジュールは 2 章で述べた見積り値管理デーモンに相当する。書き込み要求処理の効率を考慮、独立したデーモンとして実装するのではなく Xnfsd にモジュールとして組み込んだ。割当量管理テーブルは、ユーザごとにディスク割当量を保持する。見積り値管理テーブルは、ユーザごとに見積り値を保持する(図 5)。割当量は、管理者が X-NAS マネージャを介して設定する。管理者は、ユーザごとの割当量の合計値が X-NAS 全体の物理的なディスク容量(各要素 NAS のデータディスク容量の合計値)を超えないように各ユーザの割当量を設定する。rquotad は、Linux に標準で搭載されており、使用量管理デーモンに相当する。rquotad は、各要素 NAS のローカルファイルシステムにおけるユーザごとのディスク使用量情報を外部に通知する。使用量取得モジュールは、RPC(リモートプロシージャコール)を利用して各要素 NAS の rquotad にユーザごとのディスク使用量情報を問い合わせ、各要素 NAS におけるユーザごとのディスク使用量情報を取得する。そして、取得したディスク使用量情報を集計し、X-NAS 全体におけるユーザのディスク使用量を求める。各ユーザは、Linux のローカルファイルシステムのユーザ ID により管理される。

3.3 XQUOTA の処理

X-NAS におけるプラットフォーム OS である Linux のネイティブファイルシステムは、ext2¹⁸⁾ である。そこで、要素 NAS のローカルファイルシステムとして ext2 を採用した場合の、見積りベース Quota 方式の実装について説明する。まず、ext2 ファイルシステムの構造について簡単に触れた後、X-NAS におけるディスク使用量の変化量について説明する。

3.3.1 ext2 ファイルシステムの構造

図 6 に ext2 ファイルシステムのファイル構造を示す¹⁸⁾。実ファイルデータは、4KB 単位のデータブロックと間接ブロックで構成される。間接ブロックは、フ

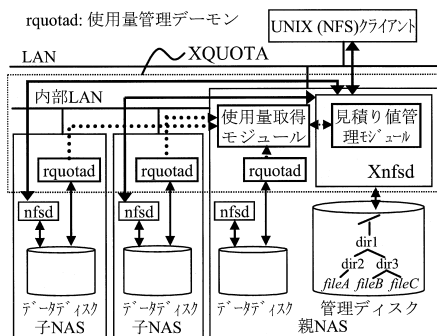


図 4 XQUOTA の構成

Fig. 4 Structure of XQUOTA.

ユーザID	122	312	...	635
割当量(KB)	4500	4500	...	6000

(a) 割当量管理テーブル

ユーザID	122	312	...	635
見積り値(KB)	1243	3432	...	234

(b) 見積り値管理テーブル

図 5 割当量管理テーブルと見積り値管理テーブルの構成

Fig. 5 Structure of tables for keeping quota limit and quota estimate.

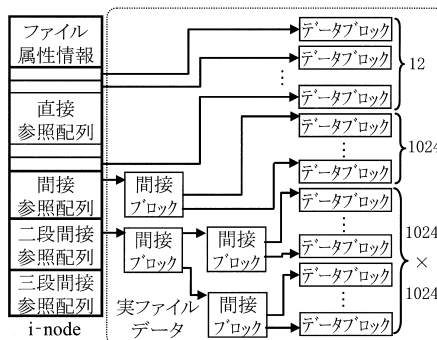


図 6 ext2 ファイルシステムのファイル構造¹⁸⁾

Fig. 6 File structure of ext2 file system.

イルサイズが大きい場合に使用される管理データであり、データブロックあるいは別の間接ブロックへのポインタを保持する。したがって、大容量ファイルの場合、データブロックに間接ブロックを加算した値が実ファイルデータのサイズとなる。

ディレクトリ構造もファイルと同様にブロックが割り当てられ、そのディレクトリ直下にあるファイルやディレクトリの管理データ(ディレクトリエントリ)が格納される。各ディレクトリエントリのサイズは数百バイト程度であるが、ディレクトリ作成と同時に 4KB ブロックが確保される。ディレクトリに保持するファイル数が増加すると、エントリ数も増加し、4KB

ブロックで管理できなくなった時点で新たに 4KB ブロックが確保される。

3.3.2 X-NAS におけるディスク使用量の変化量

見積り値は実際のディスク使用量よりも多めに見積もる必要がある。NFS プロシーダの CREATE 要求と SETATTR 要求の場合、属性にファイルサイズ情報が含まれるためその値を変化量とする。WRITE 要求と MKDIR 要求は、実ファイルデータおよび管理データを持つため、ディスク使用量が大きく変化する。そこで、X-NAS における実際のディスク使用量の変化量について調査した。

(1) WRITE 要求

X-NAS における WRITE 要求処理では、ファイルの実体を保持する特定のデータディスクに対して書き込みを行う。NFS の WRITE 要求の書き込み単位は 8KB であるため、そのデータディスクには最大 2 つのデータブロックが作成される。直接参照配列の場合、ユーザのディスク使用量は 8KB 増加する。しかし、間接参照配列の最初の要素が管理するデータブロックを作成する場合、ディスク使用量は 12KB (= 8KB (データブロック) + 4KB (間接ブロック)) 増加する。また、二段間接参照配列の最初の要素が管理するデータブロックを作成する場合、ディスク使用量は 16KB (= 8KB (データブロック) + 8KB (間接ブロック × 2)) 増加する。ext2 ファイルシステムのアーキテクチャ上は、三段間接参照もとりうるが、現状は、i ノードのサイズフィールドの制限から三段間接参照配列により管理されるデータブロックは存在しない。したがって、8KB の WRITE 要求の書き込み先 (オフセット) の差異により、ユーザのディスク使用量は最大 16KB 増加する。

(2) MKDIR 要求

X-NAS における MKDIR 要求処理では、全データディスクにディレクトリを作成する。各データディスクのローカルファイルシステムは、MKDIR 要求処理によりディレクトリエントリを 1 つ作成する。ディレクトリエントリはブロック単位で管理されるため、1 つのディレクトリエントリを作成すると 4KB 増加する。X-NAS の場合、すべてのデータディスクにディレクトリを作成するため、1 回の MKDIR 要求処理でユーザのディスク使用量は、最大 (データディスク数 × 4)KB 増加する。

```

Quota処理{
    見積り値(U_ID)+=変化量;
    if(見積り値(U_ID)>割当量(U_ID)/n){
        全要素NASからU_IDに
        対応するディスク使用量取得;
        real=Σ(各要素NASのディスク使用量);
        if(real>割当量(U_ID)){
            Quotaエラー返送;
        }else 書き込み処理;
        見積り値(U_ID)=real/n+変化量;
    }else 書き込み処理;
}
    U_ID: ユーザID
    n: 見積り値管理デーモン数
  
```

図 7 見積りベース Quota 処理アルゴリズム

Fig. 7 Algorithm for estimate-base quota processing.

3.3.3 見積りベース Quota 処理

見積りベース Quota 処理は、見積り値を更新する見積り処理と、見積り値を用いてディスク使用量が割当量に到達したか否かを判定する判定処理からなる。図 7 に見積りベース Quota 処理アルゴリズムを示す。X-NAS では、見積り値管理デーモンを親 NAS のみに搭載するため、見積り値管理デーモン数 $n = 1$ である。

見積り処理は、3.3.2 項で述べた書き込み要求処理によるディスク使用量の最大変化量を見積り値に加算する。これらの値は、1 回の書き込み要求処理によるディスク使用量の最大増加量であるため、見積り値はいかなる場合でもディスク使用量を下回らない。

判定処理は、見積り値と割当量を比較する。見積り値が割当量よりも小さい場合、Quota エラーを発行せず、ローカルファイルシステムに書き込み要求を処理させる。一方、見積り値が割当量よりも大きい場合、使用量取得モジュールによりディスク使用量情報を取得し、その値と割当量を比較する。ディスク使用量情報が割当量を下回る場合、Quota エラーを発行せず、それ以外の場合、Quota エラーと判定する。そして、見積り値と実際のディスク使用量の誤差を縮小するため、ディスク使用量を取得した後、その値で見積り値を補正する。

ただし、Quota エラーとならない場合でも、管理者の設定ミスにより、X-NAS 全体としての物理的なディスク容量に空きがなくなりデータを書き込めなくなるケースがある。このようなケースでは書き込み処理において NOSPC エラーとなり、そのエラーがクライアントに返送される。また、X-NAS 全体としての物理的なディスク容量に空きがあるが書き込み先のデータディスクに空きがないケースでは、X-NAS の自律容量リバランス機能がそのデータディスクから別のデー

タディスクへとファイルを移動して書き込み先のデータディスクに空きを作るため、書き込み処理を行うことができる。

以上のように、使用量取得処理は見積り値が割当量よりも大きい場合にのみ行うため、使用量取得処理回数を削減できる。

4. 評価

XQUOTA の有効性を検証するため、NFSv3 ベースの X-NAS プロトタイプに XQUOTA を実装し、評価した。

4.1 評価環境

表 2 に実験で使用した X-NAS 構成を示す。親 NAS には、XQUOTA を実現する各種モジュールを実装する。子 NAS は、標準的な NFS サーバであり、新規ソフトウェアを搭載しない。また、X-NAS の実用的な台数が 4 台から 8 台程度¹⁾であることを考慮し、実験では最大 8 台構成の X-NAS を使用した。

4.2 評価指標と評価方法

表 3 に、評価項目と評価指標を示す。

評価指標のうち、書き込み時間およびスループットは以下の方法で評価した。X-NAS は、クライアント PC のデータのバックアップ先として利用されることが予想される。X-NAS をバックアップ先として利用する場合、ファイルの書き込み性能が重要になる。そこで、X-NAS に一定数のファイルをまとめて書き込んだ場合の性能を測定した。ファイルセットとしては、以下の 2 つを使用した。

- ファイルセット 1

オフィスで扱うドキュメントの平均サイズ (100 ~ 300 KB) を解析用にモデル化したもの。256 KB のファイルを 900 個。

表 2 実験環境

Table 2 Experimental environment.

X-NAS の構成	親 NAS+子 NAS × 7 台
要素 NAS の仕様	OS: Red Hat 7.2 CPU: Pentium III 1GHz Memory: 1GB HDD: 36GB (Ultra160 SCSI 10000rpm) NIC: 100Mbps

表 3 評価項目と評価指標

Table 3 Performance indexes.

	評価項目	評価指標
1	管理データ規模	XQUOTA 管理情報サイズ
2	実装規模	新規開発コード量
3	耐故障性	復旧時間
4	オーバーヘッド	書き込み時間, スループット

- ファイルセット 2

著者らがレポート等のオフィスドキュメント作成に使用しているクライアント PC に保持されたドキュメント群を含む 221 MB のサイズのフォルダ。コピー対象のフォルダは 4 個のサブフォルダを含みサブフォルダには 42 個のフォルダと 335 個のファイルが格納されている。

上述のファイルセットを 1 人のユーザが cp コマンドを用いてクライアント PC から X-NAS に一度にコピーする。ユーザがファイルをコピーしている間、他ユーザは X-NAS に対してアクセスしない。

今回の評価では、XQUOTA なしの場合、XQUOTA ありの場合について、ファイルセット 1 および 2 の書き込み時間を測定した。XQUOTA ありの場合については、書き込み要求を受け付けるたびに使用量取得処理を行う問合せベース Quota 方式、見積りベース Quota 方式を使用し、ユーザの割当量を 200 MB に設定した。XQUOTA なしの場合にはファイルセットのコピー時間を書き込み時間とし、XQUOTA が有効な場合は、200 MB のファイルコピーが完了し Quota エラーが検出されるまでの時間を書き込み時間とした。

4.3 評価結果と考察

4.3.1 管理データ規模

X-NAS プロトタイプの親 NAS では、見積り値管理デーモンと使用量管理デーモンの両方を配置するため XQUOTA 管理情報のサイズが最も大きい。そこで、プロトタイプから親 NAS における XQUOTA 管理情報サイズの最大値を見積もった。X-NAS のユーザ数は最大 1,000 人であることから、親 NAS における XQUOTA の管理情報サイズは、約 6 MB [見積り値管理デーモンが管理するテーブルサイズ (最大 12 KB) + 使用量管理デーモンが管理するユーザごとのディスク使用量情報 (最大 6 MB)] となる。このサイズは、X-NAS の構成台数によらず一定である。一方、集中管理方式を採用した場合、2.3.2 項で説明したように親 NAS には要素 NAS 1 台あたり最低 160 MB の管理情報が必要であるため、8 台構成の場合には 1,280 MB になると予想される。したがって、分散管理方式により、1 台の要素 NAS に保持する XQUOTA 管理情報を 1/200 に削減できることを確認した。

4.3.2 実装規模

X-NAS プロトタイプにおける XQUOTA では、使用量の管理を各要素 NAS のローカルファイルシステムの Quota 機能と rquotad に任せ、使用量取得モジュールと見積り値管理モジュールのみを新規開発した。これらのモジュールの開発コード量は、エラー処理を含

め C コードで 500 ライン程度である．集中管理方式を採用した場合，Quota 機能に加えて構成台数の増加に対応するためのコードを新規開発する必要があるため，約 2500 ラインになると予想される．この結果から，分散管理方式の採用により，集中管理方式の約 1/5 のコード量で，クラスター型 NAS システムの Quota 機能を実現できることを確認した．

4.3.3 耐故障性

X-NAS では，故障した要素 NAS の Quota 情報を設定し直し，その情報を rquotad に問い合わせるだけで復旧することができる．X-NAS プロトタイプでは，各要素 NAS にユーザごとのディスク使用量を問い合わせ，ディスク使用量を取得するコマンドを実装している．8 台構成の X-NAS プロトタイプにおいて，そのコマンドの実行時間を測定したところ，1 秒未満であった．一方，集中管理方式を採用した場合，各要素 NAS のファイルディレクトリツリーを探索して XQUOTA 管理情報を再構成しなければならないため，(各要素 NAS における XQUOTA 管理情報作成時間 × 8) の復旧時間が必要であり秒単位での復旧は難しい．したがって，分散管理方式を採用した XQUOTA により，復旧時間を最小限にして，集中管理方式よりも耐故障性を高めることができることを確認した．

4.3.4 オーバヘッド

図 8 および図 9 に XQUOTA なし/問合せベース Quota 方式/見積りベース Quota 方式の 3 つのケースについて，X-NAS の構成台数を 1 台から 8 台まで変化させた場合のファイルセット 1 およびファイルセット 2 の書き込み時間とスループットを示す．X-NAS 台数が複数の場合，ファイルは各要素 NAS にほぼ同数ずつ書き込まれている．たとえば，3 台構成の X-NAS の各要素 NAS をそれぞれ要素 NAS1，要素 NAS2，要素 NAS3 とすると，ファイルセット 1 の 1 番目のファイルは要素 NAS1 に，2 番目のファイルは要素 NAS2 に，3 番目のファイルは要素 NAS3 にというように書き込まれ，結果として各要素 NAS に 300 個ずつ書き込まれる．

図 8 (a)，(b) とともにグラフの形状は似ているが，(b) の方が XQUOTA なしおよび見積りベース Quota 方式を用いた場合の書き込み時間が短いことが分かる．また全ケースで X-NAS の構成台数を 1 台から 2 台に増やすと書き込み時間が約 1.5～2 倍近くになっている．これは，ネットワークを経由したファイル書き込みのオーバーヘッドである．X-NAS の構成台数が 2 台以上の場合，XQUOTA なしと見積りベース Quota 方式では書き込み時間はほぼ一定である．ところが，

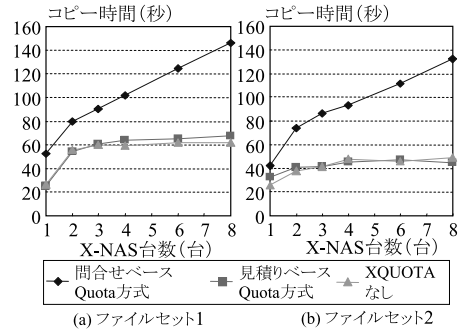


図 8 ファイル書き込み時間比較

Fig. 8 Comparison with copying times for file sets with XQUOTA.

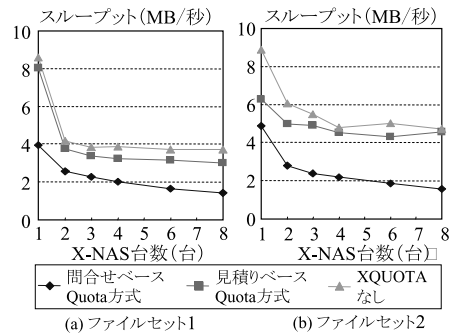


図 9 書き込みスループット比較

Fig. 9 Comparison with throughput for file sets with XQUOTA.

問合せベース Quota 方式は，XQUOTA なしケースと比べて書き込み時間が 1.5 倍から 2.5 倍になっている．そのうえ，X-NAS の構成台数が増えるにつれて XQUOTA なしのケースとの書き込み時間の差は広がっており，使用量取得処理のオーバーヘッドが非常に大きいことが分かる．一方，見積りベース Quota 方式を採用した場合の書き込み時間は，XQUOTA なしの場合に比べて平均 2.6% の増加である．図 9 に示すように，問合せベース Quota 方式の場合，構成台数を増やすに従ってスループットは低下する．一方，見積りベース Quota 方式では，構成台数が 2 台以上の場合，台数によらずスループットはほぼ一定であり XQUOTA なしの場合と同等の性能を得られる．

次に，コピー中の各書き込み要求処理の挙動を解析するため，8 台構成の X-NAS においてファイルセット 1 をコピーしたときの書き込み要求のプロファイルを取得した．図 10 に書き込み要求ごとの処理時間の推移を示す．今回の実験では，1 回のファイル書き込みにおいて，1 つの CREATE 要求と，32 個の WRITE 要求が処理される (256 KB のファイルであるため)．

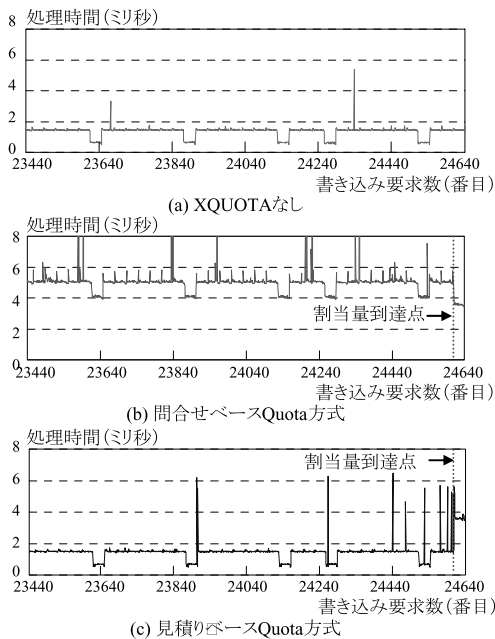


図 10 書き込み要求ごとの処理時間の推移

Fig. 10 Times for processing write operations.

複数個のファイルコピーでは、これらの要求が繰り返して処理される．図 10 の横軸は、書き込み要求処理回数であり、縦軸はクライアントからみた書き込み要求の処理時間である．図 10 (b) および (c) における図右の点線は、割当量に到達する地点を示している．XQUOTA なしの場合の書き込み要求の平均処理時間は、約 1.5 ミリ秒である（図 10 (a)）．書き込み要求の平均処理時間が 0.8 ミリ秒程度の部分は、親 NAS にファイルを書き込む場合の書き込み要求処理時間である．この差 0.7 ミリ秒が、ネットワークを経由したファイル書き込みのオーバーヘッドとなる．一方、問合せベース Quota 方式の書き込み要求の平均処理時間は、約 5 ミリ秒である（図 10 (b)）．XQUOTA なしの場合に比べて、書き込み要求の平均処理時間が 3.5 ミリ秒長い．これが使用量取得処理のオーバーヘッドである．図 10 (c) に示す見積りベース Quota 方式では、ユーザのディスク使用量が割当量よりも小さい場合は使用量取得処理を行わないため、書き込み要求の平均処理時間は、XQUOTA なしの場合とほぼ同じである．

図 10 (c) において、散発的に発生しているパルス状の突起が使用量取得処理である．見積りベース Quota 方式では、見積り値が割当量を超えた場合のみ使用量取得処理を行う．図 11 に、図 10 の見積りベース Quota 方式におけるユーザのディスク使用量の見積り値と実際のディスク使用量の推移を示す．横軸は、

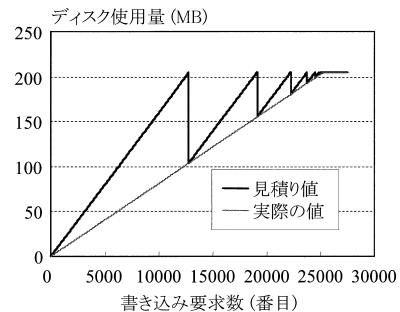


図 11 ディスク使用量の見積り値と実際の値の推移

Fig. 11 Relation between the estimate data and the real data.

表 4 XQUOTA のオーバーヘッド

Table 4 Overheads of XQUOTA.

	平均書き込み時間比
XQUOTA なし	1
問合せベース Quota 方式	1.971
見積りベース Quota 方式	1.026

図 10 と同様に書き込み要求処理回数である．図 11 では、見積り値が割当量である 200 MB に達した書き込み要求処理の直後に見積り値が実際のディスク使用量と同じになり、使用量取得処理により見積り値を補正していることが分かる．見積り値は書き込み要求処理回数（このケースでは、WRITE 要求処理回数）に比例して増加するため、使用量と割当量の差（空き容量）が小さい場合には見積り値が割当量を超えやすくなり、使用量取得処理の頻度が高くなる．

XQUOTA のオーバーヘッドは、書き込み要求の処理時間における XQUOTA に関する処理時間の占める割合である．書き込み時間は、書き込み要求処理時間の累積値であるため、書き込み時間により XQUOTA の平均オーバーヘッドを算出できる．表 4 に XQUOTA なしケースを 1 とした場合の平均書き込み時間比をまとめる．平均書き込み時間比は、X-NAS の構成台数を 1 から 8 台まで変化させ、それぞれの台数における XQUOTA なしケースを 1 とした場合の XQUOTA ありケースの書き込み時間比を求め、それらの平均をとったものである．問合せベース Quota 方式の場合、使用量取得処理によるネットワークオーバーヘッドにより XQUOTA なしケースの書き込み時間に比べて平均 97% の増加である．このオーバーヘッドは X-NAS の構成台数の増加にとまない大きくなる．一方、見積りベース Quota 方式を採用した場合の書き込み時間は、XQUOTA なしの場合の書き込み時間に比べて平均 2.6% の増加である．したがって、見積りベース Quota 方式の導入により、クラスタ型 NAS システムにおけ

る Quota 処理オーバーヘッドを平均 3%以下に削減できることを検証し、低オーバーヘッドのクラスタ型 NAS システム向けの Quota 機能を実現できることを確認した。

5. 関連研究

1 章でも述べたように、複数の NAS をクラスタ化することにより、仮想的に大容量の NAS にする技術がすでに提案されている^{6)~9)}。しかしながら、これらのクラスタ型 NAS システムには、現状 Quota 機能がサポートされていない。一方、分散ファイルシステムや分散ストレージシステムの研究では、分散環境における Quota 機能の実現方法が提案されている。

AFS^{10)~12)} は、CMU が開発した分散 UNIX ファイルシステムである。AFS は、ディスク管理を効率良く行うために、volume と呼ぶ概念を用いている。volume は、仮想的なディスクパーティションであり、管理者が効率良くディスク容量を管理できるように、多くの volume が単一の物理ディスクパーティションに割り当てられる。AFS ではユーザごとにファイルを配置するディスクを単一の物理ディスクに限定することで Quota 機能を実現する。具体的には、ユーザアカウント作成時に、そのアカウントに対して 1 つの volume が作成され、その volume に対して Quota 値が設定される。そのため、クラスタ型 NAS システムのようにユーザのファイルが複数のディスクに分散配置される場合、ユーザごとに複数のアカウントを使い分ける必要がある。

CFS¹³⁾ は、peer-to-peer の分散ストレージシステムである。各ファイルを CFS サーバと呼ぶ仮想サーバに分散して配置する。CFS は IP アドレスベースの Quota 機能をサポートする。具体的には、1 ユーザは 1 つの IP アドレスしか使用しないと仮定したうえで、各ディスクにおいて単一の IP アドレスを使用する全ユーザのディスク使用量を限定する。これにより、集中管理を行うことなく各ディスク装置において Quota 機能を実現できる。一方で、1 つの IP アドレスを複数ユーザが使用するような場合や、逆に 1 ユーザが複数のクライアント計算機を使用する場合に、ユーザごとにディスク使用量を管理するのは難しい。

PAST^{14),15)} は、CFS と同様のアプローチをとっている peer-to-peer の分散ストレージシステムである。PAST はオプションとして、サーバとクライアントの両方にスマートカードを装備する。スマートカードはセキュリティと Quota 機能の実現のために使用する。PAST では、ユーザごとのディスク使用量をクライ

アント側のスマートカードに記録することで分散ストレージシステムにおける Quota 機能を実現する。そのため、サーバだけでなくクライアント計算機にも独自の Quota 機能を搭載する必要がある。

本稿では、クラスタ型 NAS システムに適した低オーバーヘッドの Quota 機能を提案した。提案した Quota 機能は、AFS に比べてユーザ自身が複数のディスクを使い分けることなくユーザごとにディスク使用量を制限することができる。また、CFS に比べ、1 ユーザが複数のクライアント計算機を使用する場合や、1 つの IP アドレスを複数ユーザが使用する場合にも、ユーザごとにディスク使用量を制限することができる。さらに、PAST に比べクライアント計算機には特別な機能を追加することなく、ユーザごとにディスク使用量を制限することができる。本機能をクラスタ型 NAS システムに適用することにより、クラスタ型 NAS システムの使い勝手と管理容易性を向上させられる。

6. おわりに

クラスタ型 NAS システムの使い勝手と管理容易性を向上させるため、ユーザごとのディスク使用量を制限する機能 XQUOTA を開発した。XQUOTA では、管理データ規模、実装規模と耐故障性を重視し、ユーザごとのディスク使用量情報をクラスタ型 NAS システムを構成する各要素 NAS で分散して管理する分散管理方式を採用した。各要素 NAS で管理するユーザごとのディスク使用量をネットワーク経由で取得するオーバーヘッドを削減するため、NFS プロシージャの種類とローカルファイルシステムにおける実際のデータアクセスサイズを元にディスク使用量を概算し、見積り値をディスク使用量の代わりに用いる見積りベース Quota 方式を提案した。本方式の有効性を検証するため、XQUOTA を仮想一元化 NAS システム X-NAS のプロトタイプに実装し、評価実験を行った。評価の結果、XQUOTA のオーバーヘッドを削減目標 10%に対して 3%以下に抑制可能であり、クラスタ型 NAS システムに適したディスク容量制限機能を実現できることを確認した。

謝辞 本稿の執筆にあたり、匿名査読者諸氏から数多くの有益なコメントをいただいた。ここに感謝の意を表する。

参 考 文 献

- 1) Yasuda, Y., et al.: Concept and Evaluation of X-NAS: a Highly Scalable NAS System, *Proc. 20th IEEE/11th NASA Goddard Confer-*

ence on Mass Storage Systems and Technologies (MSST 2003), pp.216–224 (2003).

- 2) 川本真一ほか：ファイル自律配置方式を備えた仮想一元化 NAS システム X-NAS の実現と評価，第 14 回データ工学ワークショップ 4-B-01 (2003).
- 3) Yasuda, Y., et al.: Scalability of X-NAS: a Clustered NAS System, 情報処理学会論文誌：コンピューティングシステム, Vol.44, No.SIG11, pp.68–78 (2003).
- 4) 川本真一ほか：仮想一元化 NAS システム X-NAS における自律容量リバランス機能の実現と評価，信学技報，Vol.103, No.248, pp.1–6 (2003).
- 5) 保田淑子ほか：仮想一元化 NAS システム X-NAS における同期バックアップ機能の実現と評価，信学技報，Vol.103, No.248, pp.7–12 (2003).
- 6) Karamanolis, C., et al.: DiFFS: a Scalable Distributed File System, Technical Report HPL-2001-19, HP Laboratories Palo Alto (2001).
- 7) Karamanolis, C., et al.: An Architecture for Scalable and Manageable File Services, Technical Report HPL-2001-173, HP Laboratories Palo Alto (2001).
- 8) Anderson, D.C., et al.: Interposed Request Routing for Scalable Network Storage, *ACM Trans. Comput. Syst.*, Vol.20, No.1 (2002).
- 9) 山川 聡ほか：NAS スイッチ：NFS サーバの仮想化統合技術の開発，信学技報，CPSY2002-36 (2002).
- 10) Satyanarayanan, M.: Integrating Security in a Large Distributed System, *ACM Trans. Computer Systems*, Vol.7, No.3, pp.247–280 (1989).
- 11) Satyanarayanan, M.: Scalable, Secure, and Highly Available Distributed File Access, *COMPUTER*, pp.9–21 (1990).
- 12) Sidebotham, R.N.: Volumes; The Andrew File System Data Structuring Primitive, *Proc. European Unix User Group Conference* (1986).
- 13) Dabek, F., et al.: Wide-area cooperative storage with CFS, *Proc. SOSP'01*, pp.202–215 (2001).
- 14) Rowstron, A., et al.: Storage management and caching in PAST, a large-scale persistent peer-to-peer storage utility, *Proc. SOSP'01* (2001).
- 15) Druschel, P., et al.: PAST: A large-scale, persistent peer-to-peer storage utility, *Proc. HotOS VIII* (2001).
- 16) Callaghan, B.: *NFS Illustrated*, Addison Wesley, Reading, Massachusetts (2000).
- 17) Eckstein, R., et al.: *Using Samba*, O'Reilly and Associates, Inc (1999).
- 18) 小松克行：ext2 ファイルシステム，*Linux magazine*, 1999, No.3, pp.153–160 (1999).

(平成 16 年 1 月 30 日受付)

(平成 16 年 5 月 9 日採録)



保田 淑子 (正会員)

1991 年早稲田大学理工学部卒業。同年 (株) 日立製作所入社。入所以来中央研究所に勤務。並列計算機、スーパーコンピュータ、サーバ、ネットワークストレージの研究開発に従事。IEEE/Computer 会員。



沖津 潤

1999 年東京工業大学工学部卒業。2001 年同大学大学院修士課程修了。同年 (株) 日立製作所入社。入所以来中央研究所に勤務。ネットワークストレージ、オープンサーバの研究開発に従事。



川本 真一 (正会員)

1991 年東北大学工学部卒業。1996 年同大学大学院博士課程修了。同年同大学院助手。1998 年 (株) 日立製作所入社。入所以来中央研究所に勤務。オープンサーバ、ネットワークストレージの研究開発に従事。博士 (情報科学)。



江端 淳

1993 年筑波大学基礎工学部卒業。1995 年同大学大学院修士課程修了。同年 (株) 日立製作所入社。入所以来中央研究所に勤務。大型計算機、ネットワークストレージの研究開発に従事。



樋口 達雄

1988 年東京大学工学部卒業。1990 年同大学大学院修士課程修了。同年 (株) 日立製作所入社。入所以来中央研究所に勤務。現在、主任研究員。並列計算機、オープンサーバ、ネットワークストレージの研究開発等に従事。電気情報通信学会会員。