L-014

分散ファイルシステムにおける冗長化方式の一検討

A Study on Server Redundancy for Distributed File System

西谷 明彦† 寺下 雅人† 大岸 智彦†
Akihiko NISHITANI Masahito TERASHITA Tomohiko OGISHI

1. まえがき

クラウドサービスの普及に伴い、多くのシステムの基盤として、PC が持つストレージを統合的に利用する分散ファイルシステムの活用が着目されている。分散ファイルシステムでは、低価格な汎用 PC による構築が可能なことや、ユーザ数等の増加に応じて必要な分だけハードディスクや CPU などのリソースを増やせるスケールアウト性を有するため、設備導入コストの削減への寄与が期待される一方、数台の PC に障害が発生した場合に、保管するデータを安全に保護し、システム全体の障害を防ぐことで、サービスが停止せずに運用可能とするための可用性･信頼性も求められている。

ハードディスクなどの突発的な故障に起因するシステム障害を防ぐには、単一障害点（SPoF: Single Point of Failure）の除去が重要である。筆者らは、既存の分散ファイルシステムの仕組みを生かしつつ、単一障害点となりうるメタデータサーバを冗長化する手法を考案した。本稿ではその概要について述べる。

2. 現状の構成と課題

現状の分散ファイルシステムとして、Gfarm[1]や Lustre[2]など、ユーザが作成したファイルを保管する多数のファイルサーバ、各ファイルサーバが格納するファイルのメタデータを一元管理するメタデータサーバ、ユーザアプリケーションに対して分散ファイルシステムへのアクセス手段を提供するプロキシサーバで構成されたアーキテクチャを仮定する。この構成では、1 台のメタデータサーバが、全てのユーザからの要求を処理しデータの一貫性を維持するため、メタデータサーバ上でプロセスの異常動作などのソフトウェア障害や、ハードディスク故障などのハードウェア障害が発生した場合に、システム全体の停止を招くこととなる。これを回避するため、Act-HotStandby 構成でメタデータサーバを冗長化する手法が考えられるが、ユーザ要求処理と独立にメタデータの同期を行うため、サービスのダウンタイム発生を防ぐことが難しいという問題がある。

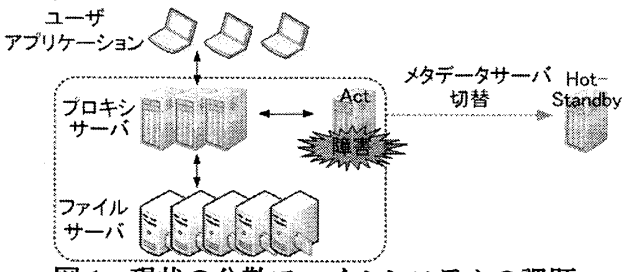

図 1 現状の分散ファイルシステムの課題

†株式会社 KDDI 研究所, KDDI R&D Laboratories, Inc.

3. 提案する冗長化方式

既存の分散ファイルシステムは、メタデータの管理や、遅延や CPU 使用率などをもとにファイルサーバを選択する仕組みなど、特徴的な機能を有する。既存の分散ファイルシステム上に冗長化機能を導入することで、既存の分散ファイルシステムの特徴を生かしつつ、課題となっているメタデータサーバの冗長化を行う方式を考案した。

3.1 アーキテクチャ

提案する冗長化方式のアーキテクチャを図 2 に示す。本構成では、既存の分散ファイルシステム相当のシステムを複数系統導入する。ユーザからの要求に対して、全ての系統が同時に動作することで、いずれかの系統で障害が発生しても他の系統にてサービスを継続する。DNS サーバは、ユーザからの問い合わせに対して、最初に要求を受け付ける系統を選択する仕組みを提供する。生存監視サーバは、各系統内のサーバの故障や復帰を迅速に検知する機能を有し、DNS 登録情報と連動させて系統別の死活を制御する役割を果たす。

なお、個々の系統内では、プロキシサーバ、メタデータサーバはそれぞれ 1 台ずつ存在し、ファイルサーバにおいては、ファイルの複製は行わない構成を想定している。

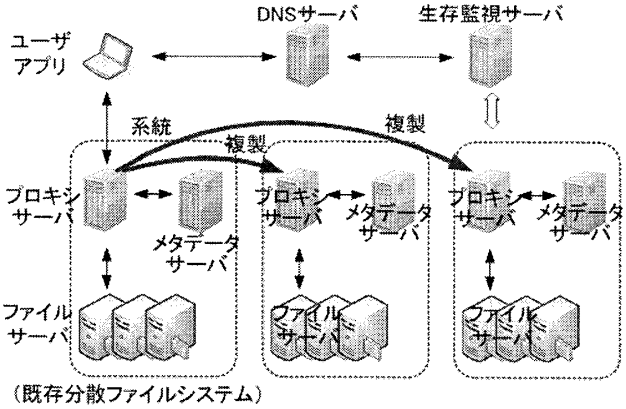

図 2 提案するアーキテクチャ

3.2 正常手順

提案方式における正常手順を以下に示す。ファイル書き込みの場合、ユーザからのファイルアクセス要求を DNS サーバが選択した一次プロキシサーバが受け取り、他の二次プロキシサーバに要求を複製することで各系統において並列に同一の要求処理を実施する。二次プロキシサーバより一定数の正常応答が得られたとき、ユーザに結果を通知する（図 3参照）。本正常応答数については、応答性能を重視する際は 1 に近い値とし、データの可用性を重視する際は系統数－1 に近い値とする。読込みの場合、一次プロキシサーバ自身での読込みが失敗した場合に限り、他の二

次プロキシサーバに読込み要求を複製し、その結果を持ってユーザ端末に応答する。

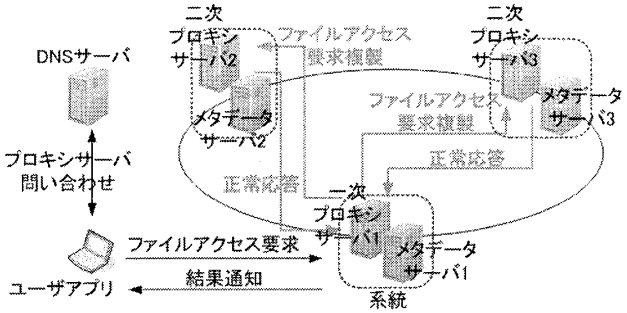

図 3　ファイルアクセス命令の複製

3.3　データの一貫性保証

ファイル書き込み要求などにおいて、いずれかの系統で失敗あるいは著しい応答遅延が発生した場合には、各系統のメタデータサーバが保持するメタデータに差分が生じる。このとき、図 4 に従いデータの一貫性保証を行う仕組みを実現する。

一次プロキシサーバは、二次プロキシサーバからエラー応答を受信した際、監視サーバにエラー報告を行う。監視サーバは、各プロキシサーバからのエラー履歴をもとに系統間でのファイルの差分を補完するため、各プロキシサーバに対し、不足しているファイルを相互に転送するように指示する。このように、系統間でのメタデータおよびファイルの同期は、ユーザの要求処理とは独立なタイミングで行う。

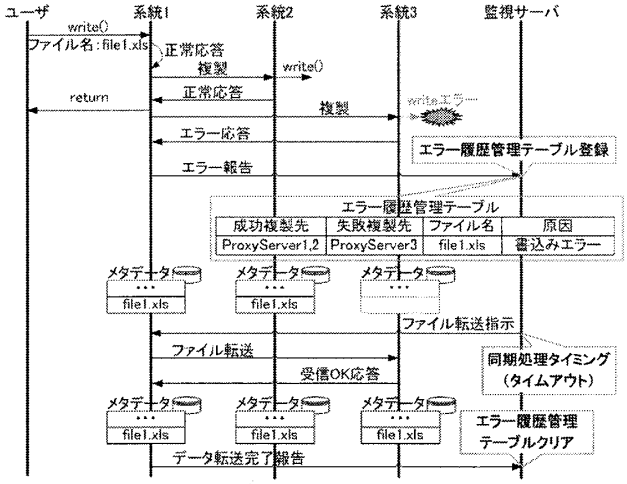

図 4　データの一貫性保証

3.4　系統別の死活監視

提案の構成において、各系統のプロキシサーバに障害がある場合、ユーザ要求の失敗という重度の障害が発生する。これを避けるため、監視サーバでは系統ごとの死活監視を行い、プロキシサーバが故障した系統を DNS サーバが選択しないような仕組みを設ける。この仕組みについて図 5 を用いて説明する。

全ての系統が正常な場合、DNS サーバは、系統 1～3 のいずれのプロキシサーバも選択する可能性がある。監視サーバは、ping およびプロセス稼働確認により、定期的に、各系統のプロキシサーバの死活監視を行う。応答が無い、あるいはプロセスが稼働していない場合、プロキシサーバの故障と判定し、DNS サーバ上にて、その系統のプロキシサーバの登録解除手続きを行う。死活監視による故障が確認できないことも考慮し、3.3 節で述べたエラー履歴も故障検知の材料として利用する。特定の系統のエラーが多い場合は、その系統の故障と判断し、同様に DNS 登録解除を行う。監視サーバは、故障した系統が正常復帰した時点で DNS サーバにその系統のプロキシサーバを再登録する。

なお、この系統の切り離し・復旧処理については故障時の対応を想定したものであり、システム構成変更による恒久的な系統数の変更については対象としていない。系統復旧後、データの一貫性保証のための同期処理を行う。

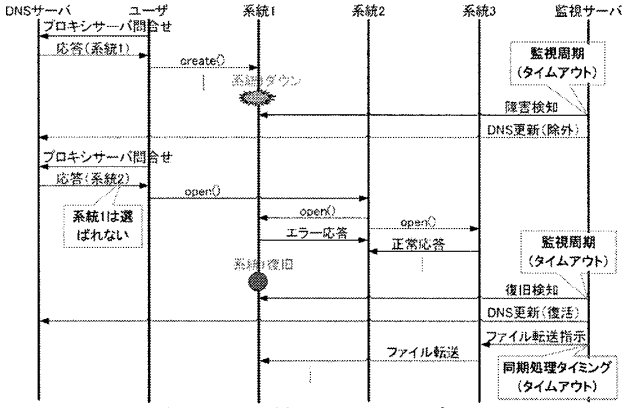

図 5　系統別の死活監視

3.5　考察

図 2では、プロキシサーバ、メタデータサーバが、それぞれ別々のマシンで動作する例を示したが、サーバプログラムを 1 台のマシン上で動作させることによりリソースの効率的利用が可能である。また、ファイルサーバについては、異なる系統のファイルサーバを同一マシン上で動作することも可能であり、これによりディスク領域の効率的利用にも貢献できると考えられる。

一方で、今回の設計では、ユーザ数が増えた場合に、プロキシサーバ／メタデータサーバの処理能力を維持するために系統数を増加させる必要がある。全ての系統が並列に動作することを前提としているため、スケーラビリティに課題が残る。

4. まとめ

本稿では、既存の分散ファイルシステムにおける、メタデータサーバが単一障害点となり得る問題に対し、その対策としての冗長化手法と設計について述べた。今後の課題として、スケーラビリティの向上が挙げられる。

謝辞

日頃よりご指導頂く、株式会社 KDDI 研究所秋葉所長、中島副所長、及び鈴木取締役に感謝致します。

参考文献

[1] 建部 修見，曽田 哲之，"広域分散ファイルシステム Gfarm v2 の実装と評価" 情報処理学会研究報告，2007-HPC-113, 2007.

[2] http://wiki.lustre.org/index.php/Main_Page