

分散ストレージにおけるパリティに基づいたデータ保護方式のデータ一貫性検証方法 Data Consistency Verification of Parity Based Data Protection for a Distributed Storage

江原 寛人[†]
Hiroto Ebara

山本 貴大[†]
Takahiro Yamamoto

坏 弘明[†]
Hiroaki Akutsu

揚妻 匡邦[†]
Masakuni Agetsuma

1. はじめに

近年、汎用サーバを用いてストレージシステムを構築する SDS (Software Defined Storage) が普及している。SDS とは、SSD (Solid State Drive) 等のデータ記憶装置 (以降、ドライブと呼ぶ) を搭載する複数の汎用サーバ (以降、ノードと呼ぶ) を Ethernet 等のネットワークを介して接続し、ストレージ機能を提供する分散ストレージである。

分散ストレージは、ノードやドライブ障害発生時のデータ消失を防止するため、ノード間でデータを冗長化して保護する。データ保護を行う方式として、単純にノード間でデータのコピーを保持する方式や、複数のノードに格納するデータ同士で冗長符号 (以降、パリティと呼ぶ) を生成し、パリティを別ノードに格納することで容量効率を向上したデータ保護方式がある。障害時には、各ノードのデータやパリティを使いデータを回復するため、ノード間でデータ不整合が起きないように設計する必要がある。

本稿では、分散ストレージにおけるパリティに基づいたデータ保護方式が、ドライブやノード障害起因のあらゆるシステム状態に対して、ノード間で常にデータ一貫性を保つ設計になっているか形式手法を用いて検証する方法について述べる。

2. 目的と課題

本研究は、分散ストレージにおいてパリティに基づいたデータ保護方式の設計を検証対象とし、データ一貫性を要件として無矛盾性の検証を目的とする。検証方法は、形式手法の一種であるモデル検査を用いる。モデル検査は、検証対象の設計と満たすべき要件をモデル記述言語で記述し、モデルが遷移可能な状態の中で要件を満たさない状態が存在するかを網羅的に検証する手法である。

本検証の課題は、検証対象の状態数が多く、検証が完了しないことである。モデル検査は、状態数の増加に伴い、検証に必要な CPU 演算量とメモリ量が著しく増加するため、検証が完了しないことがある。本検証対象の状態数が多くなる理由を以下に示す。

(理由 1) システムを構成するドライブ数およびノード数が増えると Read/Write 処理対象のデータ数が増加する。

(理由 2) 本システムは、ホストから受領した複数の Read/Write 処理を並列実行する。この時、各 Read/Write 処理は、任意の実行時点で中断され、別の処理が実行され得る。このため、Read/Write 処理の処理順序は一意に決まらず、実行する度に異なる状態遷移を辿り得る。

3. 検証方法

3.1 システム構成

検証対象のシステム構成を図 1(a)に示す。本検証対象は、5 台のノードとノード毎 1 台以上のドライブで構成し、格納データに対してデータ保護方式として MEC (Multi-stage

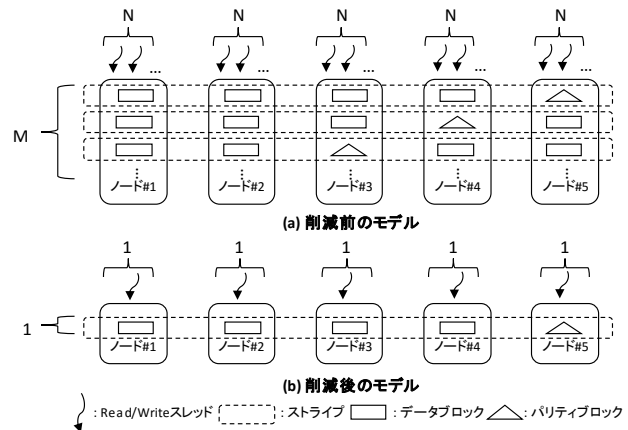


図 1 検証対象のシステム構成

Erasure Coding) 方式[1]を適用する。各ノードのドライブは、データとパリティを格納しており、これらデータとパリティは、同サイズのブロックと呼ばれる単位で区切られる。さらに、異なるノードに格納したデータブロックとパリティブロックで組を作り、パリティブロックは同じ組のデータブロックから生成する。この組をストライプと呼ぶ。

各ノードは、ホストから任意の契機で任意の数の Read 及び Write 要求を受領し、Read/Write スレッドを実行する。Read スレッドは、要求されたアドレスからデータを読み出して応答する。読み出し先が障害状態である場合、パリティブロックと他のデータブロックから読み出し先のデータを復号して応答する。Write スレッドは、受信したデータを要求されたアドレスに書き込み、さらに書き込み先のデータブロックに対応するパリティブロックを更新して応答する。MEC 方式は、一般的なパリティによるデータ保護方式[3]と比較し、データブロックを各ノードで独立して読み出し可能な特徴を有するため、Read/Write 処理は、単一ストライプ内のデータブロック単位で独立に排他制御を行う。

本検証対象は、ストライプのデータブロック数を 4、パリティブロック数を 1、システム内のストライプ数を 1000、Read/Write スレッド数をノード毎最大 32 とする。

ここで、本検証におけるデータ一貫性とは、「単一のストライプについて、当該ストライプに属する全データブロックを XOR 演算した値がパリティブロックの値と一致すること。また、障害状態にあるデータブロックは、当該データブロックの最後の更新値と当該ストライプに属する他のデータブロックとパリティブロックを XOR 演算した値が一致すること。」と定義する。この要件について、Read 時にデータ一貫性が満たされていれば、変質したデータを応答することがないため、Read 時に満たされていればよい。

[†](株)日立製作所 研究開発グループ デジタルテクノロジーイノベーションセンタ, Center for Technology Innovation - Digital Technology, Research & Development Group, Hitachi, Ltd.

3.2 モデルの状態数削減

モデルの状態数を削減するアプローチとして、モデルの変数の削減とモデルのスレッドの削減の2つがある。本検証対象のシステムにおいては、変数の削減としてストライプ数の削減、及びスレッドの削減として Read/Write スレッド数の削減がそれぞれ適用できる。

本検証では、ストライプ数と Read/Write スレッド数をそれぞれ1としたモデルの検証が、前述した検証対象のシステム構成のモデルの検証と等価であること証明することで、モデルの状態数を削減する。

（命題）3.1節で示したシステム構成において、ストライプ数を M 、ノードあたりの Read/Write スレッド数を N （ M, N は1以上の自然数）とした時、 $M=1$ かつ $N=1$ のシステム構成がデータ一貫性を満たすならば、任意の M かつ任意の N のシステム構成がデータ一貫性を満たす。

（証明）データ一貫性の定義より、データ一貫性は単一のストライプに関する性質である。ゆえに、複数のストライプが存在するシステムにおいて、異なるストライプ間の状態は互いにデータ一貫性に関して影響を与えない。したがって、「単一のストライプについてデータ一貫性が満たされるならば、任意の数のストライプについてデータ一貫性が満たされる(*)」といえる。

次に、(*)に加えて3.1節のシステム構成における排他制御のため、ストライプ数が1の場合、当該ストライプに属する各データブロックに対して Read/Write 処理を同時に実行できるスレッド数は常に1となる。ゆえに、複数のスレッドが存在したとしても、同時には1スレッドしか実行しないため、各ノードにてスレッド数が1の場合について、一貫性が満たされるならば、各ノードにてスレッド数が任意の数の場合でも一貫性が満たされるといえる。

以上より、（命題）は成り立つといえる。

3.3 モデル化

ストライプ数が1かつノードあたりの Read/Write スレッド数が1の場合の検証対象システムの構成（図1(b)）をモデル検査器 SPIN[2]のモデル記述言語 Promelaによりモデル化する。本モデルは、状態変数としてストライプとデータブロックとパリティブロックに対応するロックを有し、状態を変更する手続きとして Read/Write スレッドを有する。

Read/Write スレッドの処理フローを図2に示す通りである。また、ホストからの Read/Write 要求をモデル化するため、Promelaが有する非決定性構文を用いて Read/Write 処理を非決定的に実行するようにした。

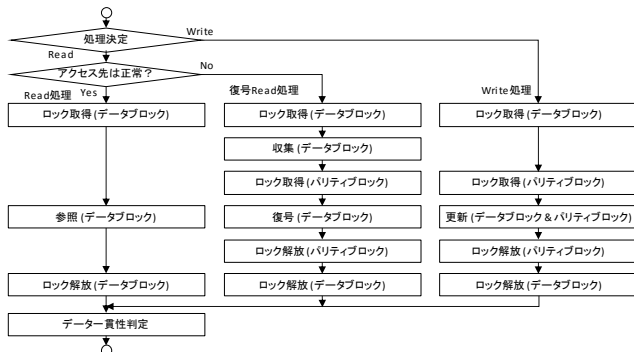


図2 Read/Write スレッドの処理フロー

モデル検査は、Read スレッドの実行が完了した時にデータ一貫性の定義を判定条件として評価し、偽の状態が存在する場合、要件を満たさないとした。

4. 評価

検証対象システムの構成を設計通りモデル化した場合（削減前）と、検証対象システムの構成を3.2節の考え方にに基づきモデル化した場合（削減後）を SPIN により実行した結果を示す。評価項目は、SPINの実行時間、及び実行結果より得られる状態数と使用メモリ量の3項目とした。

状態数は、SPINが実際に検査できた状態数を表す。使用メモリ量が評価環境上で SPIN が使用可能な値を超えた場合、実行時間が著しく増加する。このため、本評価では、使用可能な値に到達した時点で検査を停止し、結果を取得した。

評価環境を表1に示す。メモリ16GiBのうち SPIN が使用可能なメモリ量は最大約14GiBである。

パラメータと実行結果を表2に示す。削減前のストライプ数は評価環境において検証用実行バイナリを作成可能な値に制限した。削減前は120秒程度実行した時点でメモリ使用量が使用可能な値（14GiB）まで到達したため、停止させた。停止時点の状態数は5900万であった。一方、削減後は SPIN の実行が約6秒程度で完了し、状態数は約111万であった。また、削減後について、出力結果よりモデルに矛盾が無いことを確認した。以上より、検証対象がデータ一貫性を満たすことを確認できた。

表1 評価環境

CPU	Intel(R)Core(TM) i7-8700 3.2GHz 6 Core
メモリ	16 GiB
ソフトウェア	SPIN 6.4.9

表2 パラメータと実行結果

		削減前	削減後
パラメータ	ストライプ数	1000	1
	ノード毎スレッド数	32	1
実行結果	実行時間	120秒以上	約6秒
	状態数	5900万以上	約111万
	メモリ使用量	14GiB以上	約1GiB

5. おわりに

本稿では、分散ストレージにおけるパリティに基づいたデータ保護方式のデータ一貫性を、形式手法の一種であるモデル検査を用いて検証した。検証対象システムの状態数が多く、検証が完了しない課題に対して、状態数を削減してモデル化する方法を提案した。また、状態数を削減してモデル化した検証結果が、削減前の検証結果と等価であることを示した。評価より、検証対象がデータ一貫性を満たすことを確認した。今後の課題として、パリティ数を拡張した場合の状態数削減方法の検討がある。

参考文献

- [1] Hiroaki Akutsu, Takahiro Yamamoto, Kazunori Ueda, Hideo Saito, "MEC: Network Optimized Multi-stage Erasure Coding for Scalable Storage Systems", IEEE 22nd Pacific Rim International Symposium on Dependable Computing (2017)
- [2] Gerard J. Holzmann, "The Model Checker SPIN", IEEE Transactions on Software Engineering Vol.23, No.5 (1997), pp.279-295
- [3] M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, D. Borthakur, "XORing elephants: Novel erasure codes for big data", Proceedings of the VLDB Endowment (PVLDB), vol.6, no.5 (2013)