7A-04

消失訂正符号を適用した分散ストレージ向け データローカリティを意識したクラスタ構成変更方式

山本　貴大† 　江原　寛人† 　千葉　武尊† 　揚妻　匡邦†
(株)日立製作所　研究開発グループ†

1. はじめに

近年，スモールスタートでき，必要に応じてシステムを拡張可能なスケーラビリティを有した IT インフラストラクチャとして，分散ストレージが普及してきている[1].

分散ストレージは，外部記憶装置（以降，ドライブ）を搭載する複数の汎用サーバをネットワークで接続し，ストレージシステムを構築する．分散ストレージは，サーバ障害が生じても入出力処理を継続するため，サーバ間に跨りデータと冗長データを格納する．データの冗長化方式には，レプリカ方式とパリティ方式がある．レプリカ方式は，書き込むデータの複製を別サーバに格納する．このため，利用可能な容量が 1/(1+p)（p は冗長度）となり，容量効率が低い．一方，パリティ方式は，容量効率の点でレプリカ方式よりも優位であり，パリティ方式は，書き込むデータを複数組み合わせて消失訂正符号（パリティ）を算出し，パリティをデータとは別サーバに格納する．

また分散ストレージでは，高性能化のため，データはネットワークを介さずに，入出力要求を受信したサーバに搭載されたドライブ（以降，ローカルドライブ）から読み出せることが望ましい．これをデータローカリティと呼び，**図 1** に示すように分散ストレージにおいてデータローカリティを維持しつつ，パリティ方式でデータ保護する方式が提案されている[2].

また分散ストレージは，必要に応じてクラスタにサーバを増設する運用が想定され，その時に性能影響をできる限り与えることなく処理できることが望ましい．

本稿では，消失訂正符号を適用した分散ストレージにおいてデータローカリティを維持したままサーバ増設する方式の確立を目的として検討した結果を述べる．検討に際し，サーバ増設中の入出力性能低下率は，増設処理用に割り当てたネットワーク帯域にのみ依存し，10%を割り当てた時，性能低下率が 10%となることを目標とした．

2. 従来手法

分散ストレージは，サーバを増設した時，データとパリティを再配置し，容量と入出力負荷（つまり性能）をサーバ間で均一化することで，クラスタの状態を最適化する．

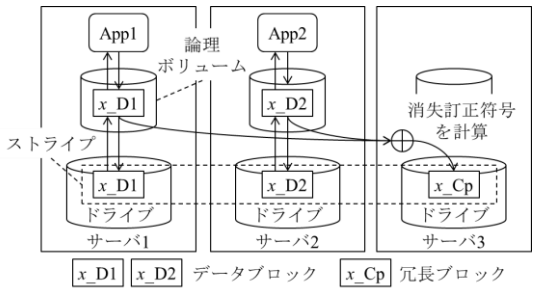

図 1 データローカリティを有したデータ保護方式

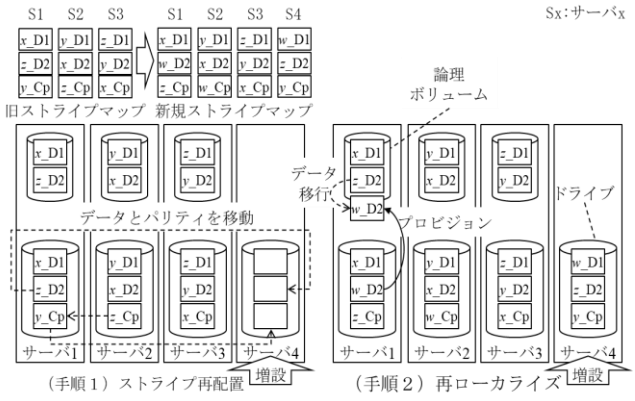

図 2　従来手法

容量と性能の観点で，データとパリティを複数のストレージコンポーネントに効率的に再配置する技術は，従来より分散 RAID 技術の分野で研究されている[3]．データローカリティを有した分散ストレージに対して，この技術を適用した場合のサーバ増設手順を説明する．

（手順１）ストライプ再配置

ストライプとはデータとそのパリティの組み合わせである．ストライプ再配置は，増設した容量を使い切るようなストライプの配置を決定するマップ（以降，ストライプマップ）を作成する．図 2 の旧ストライプマップは，増設前の 3 サーバでのストライプの配置を示しており，S1～S3 は，サーバ 1～サーバ 3 に対応する．新規ストライプマップは増設後の 4 サーバでのストライプの配置を示している．

次に新規ストライプマップに合わせてデータとパリティをサーバ間で移動する．図 2 では，配置が変更されている z_D2 をサーバ 1 からサーバ 4 に移動している．*y*_Cp と z_Cp も同様に移動する．移動元になった物理領域は，新規ストライプ（*w*_D1，*w*_D2，*w*_Cp）を割り当てて使用する．

（手順２）再ローカライズ

ストライプの再配置により，データローカリティを消失したデータ（z_D2）について，入出力要求元のサーバの空き領域（*w*_D2）にデータを移動する．この時，**図 2** に示すように論理ボリューム層でデータを移動することで，ストライプの関係は崩さずにデータの内容だけを移動する．移動後，移動元になった領域（z_D2）は，サーバ 1 から解放し，サーバ 4 に割り当てて領域を使用する．

3. 課題

従来手法は，サーバ増設中にデータがサーバ間を跨って

Data Locality Aware Cluster Reconfiguration for Erasure Coding based Distributed Storage
Takahiro Yamamoto†, Hiroto Ebara†, Takeru Chiba†, Masakuni Agetsuma†
† Research & Development Group, Hitachi, Ltd.

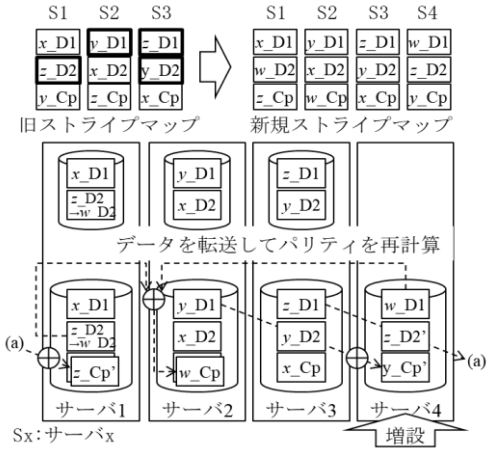

図 3　提案手法

移動するため，データローカリティを一時的に消失し，入出力処理において追加でデータのネットワーク転送が必要になる．このため，サーバ増設中に分散ストレージの入出力性能が低下する．また，（手順１）後に（手順２）が必要なため，サーバ増設完了までの処理時間が長期化する．

以上からサーバ増設中における性能低下の防止，およびサーバ増設完了までの処理時間の削減が課題となる．

4. 提案手法

上述の課題に対し，データの配置を変えずにパリティを再計算することでストライプを再配置する手法を提案する．提案手法では，まず従来手法と同様に作成した新規のストライプマップに基づいて，ストライプの組み合わせが変更されているデータを特定する．次に当該データをローカルドライブから読み出し，新規の冗長化先サーバに転送してパリティを再計算する．

図 3 を用いて具体的に説明する．**図 3** は，**図 2** と同様に 3 サーバから 4 サーバに増設した時の動作概要を示しており，図 2 と同じストライプマップを適用する．まずストライプの組み合わせが変更されているデータとして *y*_D1，*y*_D2，*z*_D1，*z*_D2 を特定する．次に各データについてローカルドライブからデータを読み出し，新規ストライプマップに合わせてデータを転送することでパリティを再計算する．*y*_D1 と *y*_D2 は，サーバ 4 にパリティ *y*_Cp’を再計算する．*z*_D1 は，サーバ 4 の領域 *z*_D2’と組み合わせてサーバ 1 にパリティ *z*_Cp’を再計算する．最後に *z*_D2 は，新規ストライプ *w*_D2 として，サーバ 4 の領域 *w*_D1 と組み合わせてサーバ 2 にパリティ *w*_Cp を再計算する．

以上から提案手法によれば，データ配置をサーバ間で変更することなく，ストライプを再配置できるため，データローカリティを維持したままサーバ増設することができる．

表 1 前提条件

#	項目	値
1	サーバ台数（＋増設台数）	5 (+1) 台
2	ネットワーク構成	10Gbps/server
3	クラスタ再構築に利用可能なネットワーク帯域割合	10%
4	データ保護設定*	4D1P

* xDyP: x 個のデータから y 個のパリティを生成

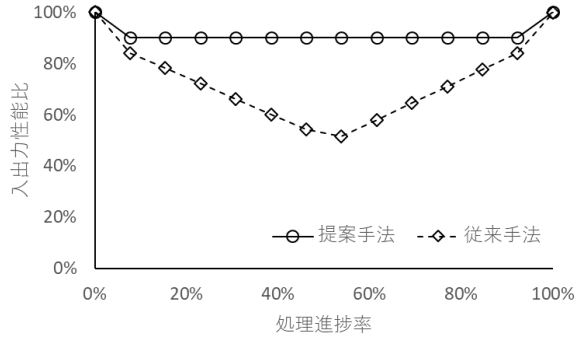

図 4　入出力処理性能の影響

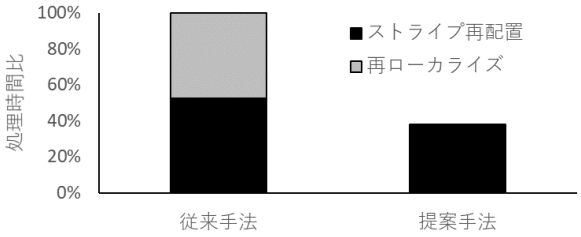

図 5　処理時間の比較

5. 評価

従来技術と提案方式について，入出力性能の低下率と構成変更時間の観点で机上評価した．評価の前提条件を表 1 に示す．なお，本評価では，ネットワーク帯域が性能のボトルネックとなる場合を仮定して評価した．

入出力性能への影響の評価結果を**図 4** に示す．**図 4** は，サーバ増設前の性能を 100%とした時の増設中の性能を比で示す．**図 4** より，従来手法はサーバ増設後，再配置されたデータが増加するに伴いデータローカリティを消失する領域が増えるため，入出力性能が最大で 50%まで低下する．進捗率が 50%を経過した時点でストライプ再配置が完了し，これ以降はデータローカリティの回復に伴い性能も元に戻っている．一方，提案手法は，サーバ増設中も常にデータローカリティがあるため，再配置に使用するネットワーク帯域分しか性能影響を受けず低下率は 10%となる．

次に構成変更時間の評価結果を**図 5** に示す．**図 5** より，提案手法は，従来手法と比べ処理完了までの時間が 60%短い．これはデータローカリティを元に戻すためにデータを書き直す（手順２）が不要となるためである．

6. おわりに

本稿では，消失訂正符号を適用した分散ストレージにおいて，データローカリティを維持したままサーバを増設する方式を提案した．評価の結果，サーバ増設中の性能低下率は 10%であり，目標を達成することを示した．

参考文献

[1] IDC: Worldwide Software-Defined Storage Forecast, 2018–2022 (online), available from <https://www.idc.com/getdoc.jsp?containerId=US44519218> (accessed 2020-01-07).

[2] Akutsu, Hiroaki, et al. “MEC: Network Optimized Multi-stage Erasure Coding for Scalable Storage Systems.” 2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC). IEEE, 2017.

[3] Holland, Mark. “On-Line Data Reconstruction in Redundant Disk Arrays.” No.CMU-CS-94-164. Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, 1994.