C-004

ドライブ増設を考慮した分散 RAID のデータ配置手法の提案と評価
Proposal and Evaluation of a Data Placement Method for Distributed RAID Considering Drive Expansion

藤井 裕大† 千葉 武尊† 大平 良徳† 出口 彰†
Hiroki Fujii Takeru Chiba Yoshinori Ohira Akira Deguchi

1. はじめに

ストレージシステムでのデータ保護技術として、従来より RAID[1]が利用されている。RAID は、データと、データから生成したパリティデータを異なるドライブに配置することで冗長性を確保する技術である。しかし近年、フラッシュメモリの微細化の進展により [2]、ストレージシステムに搭載される SSD が大容量化し、ドライブ故障時のデータ再構築（リビルド）が長期化するという問題が生じている。リビルド時間の長期化は、リビルド中のドライブ多重故障の危険性を増大させ、信頼性の低下や性能低下の長期化を引き起こす。また、大容量化により RAID グループ単位での容量設計では過剰となり、1 台単位の容量設計が求められている。

1.1 分散 RAID

リビルド時間の長期化という問題を解決する方式として、分散 RAID がある。分散 RAID は、任意の n 台のドライブに RAID 幅（データとパリティのブロック数の和）が k（≦n）のストライプをマップすることで、リビルド時のドライブ負荷を分散し、リビルド時間を短縮する(Figure1)。リビルド速度を高速化するためには、ドライブ間の Read 負荷が均一であること、即ちデータの分散度が高いことが望ましい。

1.2 分散 RAID におけるデータ配置手法と課題

分散度の高いデータ配置(データマップ)について、Balanced Incomplete Block Design(BIBD)を使った方式が提案されている[3][4]。BIBD は当該問題を一般化した組合せ問題であり、多くの分野で応用されている。しかし、現在までに一般解は発見されておらず、条件によっては BIBD を構成できないことも判明している。すなわち、任意のドライブ台数に対応することが困難であり、拡張性に乏しいと考えられる。

準最適な解法として、任意のドライブ台数 n に対する準最適マップを効率的に作成する方法が提案されており、高分散効率を実現している[5]。しかしながら当該方式では、ドライブ台数が異なるマップ間で配置差分が大きく、運用中にドライブ数を変更する際に大量のデータ移動が必要となる(Figure2)。データ移動負荷が大きくなることで I/O 性能低下や、増設処理完了までの時間が長くなるという問題がある。そのため、データ配置手法においては、データ分散度の高さと共に、ドライブ増設時のデータ移動負荷が小さい、すなわちデータ移動量が少ないことも指標として求められる。

† 株式会社日立製作所 Hitachi, Ltd.

2. データ移動量最小なデータマッピングの提案

本稿では、ドライブ増設時のデータ移動量を最小化するデータマッピング手法を提案する。提案手法では、ドライブ n-1 台のマップを基にして、配置差分が小さくなるよう n 台構成のマップを作成する。これにより、ドライブ増設時のデータ移動量の最小化を行う。またこの制約条件のもと分散度の高いデータ配置を実現するため、乱数を用いて作成したデータマッピングを複数個作成し、最も分散度の高いマップを採用する。

2.1 基本方式

分散 RAID を構成するドライブ数を n とする。ドライブ内を r 個のブロックに分割し、このブロックをストライプに k 個ずつ割り当てる(マッピング)。ストライプの数は r・n / k である。任意のドライブ数 n で効率的にストライプを定義するために、r は k の倍数(r=A・k)とする。提案マッピングの作成手順を以下に示す。

(1) n = k の場合：すべてのドライブから 1 つずつブロックを選択し、ストライプに割り当てる。

(2) n > k の場合：n - 1 台構成のマップを使い、A (n - 1) 個のストライプのマップを仮決定する。増設により新規に定義される A 個のストライプの構成ブロックを以下の手順で割り当て、マップを作成する。

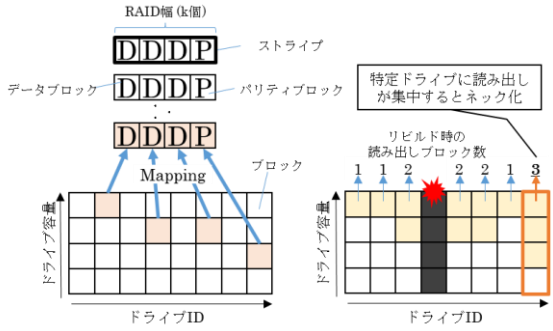

Figure 1 分散 RAID のデータ配置とリビルド速度

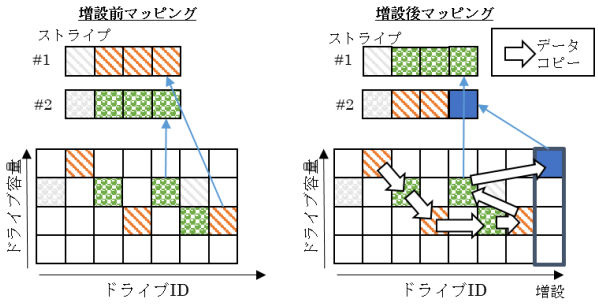

Figure 2 ドライブ増設時のデータ移動

Step1: 増設ドライブから 1 ブロックを選択し割り当てる。
Step2: 既存のストライプから 1 つを選び、そのストライプを構成するブロックから 1 つを選択して新ストライプに割り当てる。割り当てるブロック、以下の 2 つの条件を満たすブロックからランダムに選択する。
条件 1:ブロック選択回数が最小のドライブから選択
条件 2:新ストライプ内で既に割り当てられたブロックと、所属ドライブが重複しない (RAID の冗長化条件を満たす)
Step3:最後に、割当ブロックの代わりに、既存ストライプに増設ドライブから 1 ブロックを割り当てる。
Step4: Step2,3 を繰り返し k 個のブロックを割り当てる
以上の手順により、必ず増設ドライブ内のブロックがデータ移動先となるような新規マップが定義される。

2.2 複数回試行によるデータ分散度向上

提案方式において、移動対象ブロックの選択候補は無数にあり、選択対象によって最終的なマップが変化する。そのため、同一構成のマップ作成を乱数を変えて複数回繰り返し、最も分散度の高くなるマップを選択する。

3. 評価

提案方式を、増設時のデータ移動量およびリビルド効率(データ分散度)により評価する。RAID 幅(k)は 8 とした(e.g. 6 データブロック、2 パリティブロック)。

3.1 増設時のデータ移動量

n-1 台から n 台のドライブ構成に変更する際のデータ移動量について Figure4 に示す。従来方式では、ドライブ台数間で相関がなく、また r が n により変化するためデータアラインも変化するため、既存データの全量移動が必要となる。提案手法では、ドライブ台数間の相関があることと、r が一定であるため、データ移動量がドライブ台数によらず一定(ドライブ容量の 1-1/k 倍)となる。

3.2 リビルド効率の評価

リビルド効率を表す指標としてデータ分散度 α を定義する。ここで、λ_{avg} は任意のドライブに障害が発生した場合に読み込む各ドライブの平均データ量(ブロック数)、λ_{max} は最大データ量である。α は理想的にデータ分散した配置に対するリビルド速度の倍率を示し、理想的なデータ分散では 1 となる。

$$\alpha = \frac{\lambda_{avg}}{\lambda_{max}}$$

Figure5 に、マップ作成の試行回数(t)ごとのデータ分散度を示す。試行回数を増やすと分散度が上昇し、リビルド速度が増加することがわかる。100000 回の試行を行ったマップは、64 台構成において、試行回数 1 回のマップに対して 0.06 程度向上し、リビルド効率では 10.4%向上する。次に、準最適配置である RELPR[3]の分散効率と比較する。RELPR は構成台数によって分散効率が大きく増減し、提案手法は単調減少傾向である。少ないドライブ台数においては、提案手法のリビルド効率が優位と考えられる。

4. 結論

本研究では、分散 RAID のデータ配置手法として、ドライブ増設時のデータ移動量を最小化する制約のもと、高いデータ分散度を実現する配置手法を提案した。提案方式は、増設前のデータ配置を基にして、配置差分を最小化するよう増設後のデータ配置を決定する。このアプローチによりドライブ増設時のデータ移動負荷が軽減される。また、実験結果から、試行回数を増やすことでデータ分散度が向上することを確認した。

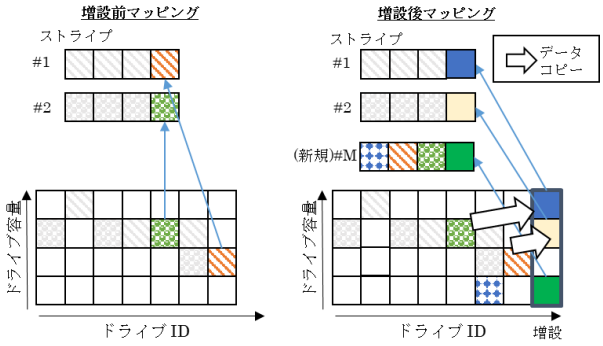

Figure 3 提案データマッピング手法の概要

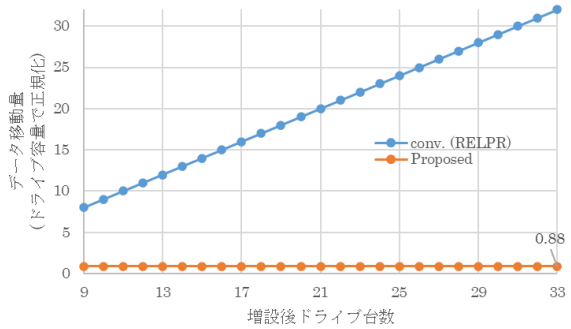

Figure 4 ドライブ 1 台増設時のデータ移動量

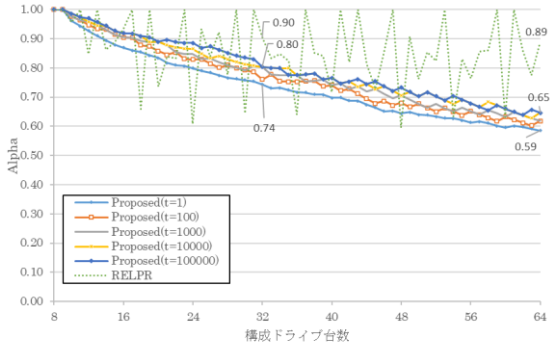

Figure 5 分散効率の評価

参考文献

[1] D. A. Patterson, et. al. , “A Case for Redundant Arrays of Inexpensive Disks (RAID)”, ACM SIGMOD, (1988).
[2] S. I. Shim, et. al., “Trends and Future Challenges of 3D NAND Flash Memory”, 2023 IEEE International Memory Workshop (IMW), (2023).
[3] M. Holland, et. al. , “Parity Declustering for Continuous Operation in Redundant Disk Arrays,” ACM SIGPLAN Notices, (1992).
[4] G. Zhang, et. al. , “RAID+: Deterministic and Balanced Data Distribution for Large Disk Enclosures”, 16th USENIX Conference on File and Storage Technologies, (2018).
[5] G. A. Alverez, et. al. , “Declustered disk array architectures with optimal and near-optimal parallelism” , Proceedings. 25th Annual International Symposium on Computer Architecture, (1998).