Vol.2014-ARC-213 No.14
Vol.2014-HPC-147 No.14
2014/12/9

Cluster-wide RAID向けの集中型コントローラ

大辻 弘貴[1,3,4]　建部 修見[2,3]

概要：高速かつ高信頼な分散ストレージシステムは，ビッグデータ処理やデータインテンシブコンピューティングを支える重要な存在である．筆者らはこれまで，分散ストレージシステム上において RAID に似た構成（Cluster-wide RAID）を構築し，ノードレベルの冗長性を確保する方法ととの性能を向上する方法について研究を進めてきた．これまでは，ストレージノード間における分散型の処理を行うことにより性能を高めていたが，データ処理能力を有するネットワークスイッチなどが登場する可能性を考慮し，集中型のアプローチにも取り組み始めたところである．本稿では，集中型の Cluster-wide RAID コントローラについてその実装と予備評価，分散型のアプローチとの比較を示す．

1. 序論

今後数年で，エクサスケールコンピューティングが実現すると見込まれている．同時に，分散ストレージシステムに要求される要件も高まると考えられ，必要な要素技術の研究開発は急務である．特に，高速かつ高信頼な大容量データの取り扱いは重要であり，ネットワークや記憶装置の性能を最大限に引き出せるメカニズムが必要である．

分散ストレージシステムにおいて耐故障性を確保するための取り組みとして，ファイル複製 [1] が多く用いられている．しかしながら，データを複製する以上は最低２倍の記憶領域が必要となり，結果的にシステムのコストを引き上げてしまう．また，複製過程において，多くのネットワーク帯域を消費してしまう問題もある．一方で，分散ストレージでない場合に関しては，従来から RAID[2] が用いられてきた．当然ながらこれは単一ノード内における記憶装置の冗長性確保に留まり，ノードレベルの耐故障性には別の方法が必要となる．本研究がターゲットとする Cluster-wide RAID は，ネットワーク上に RAID に似た構造を展開するものである．従来このようなアプローチは十分な性能が得られないとされていたが，筆者らのこれまでの研究 [3] において，高速なネットワークと RDMA(遠隔メモリアクセス) を最大限活用することで実用性の高いシステムを構築できる可能性が示された．

本稿においては，筆者らがこれまで取り組んできたノードレベルにおける耐故障性を備える Cluster-wide RAID に関して，ネットワーク上にコントローラノードを配置し，その上で冗長符号の生成などを行い，性能低下を最小化あるいは排除する方法について提案する．

1 筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba
2 筑波大学システム情報系
Faculty of Engineering, Information and Systems, University of Tsukuba
3 独立行政法人科学技術振興機構 CREST
JST CREST
4 独立行政法人日本学術振興会 特別研究員 (DC2)
JSPS DC2

2. 関連研究

本研究は，分散環境上に冗長性を持ったストレージシステムを構築することから，関連研究とては分散ファイルシステムや冗長符号に関するものが挙げられる．よく知られた分散ファイルシステムとしては，Gfarm [4]，Lustre[5]，HDFS [6][7] が挙げられる．Gfarm は複製機能を備え，HDFS については複製に加えて Erasure Coding を導入する試みもなされている [8]．いずれも複製をサポートするものであるか，Erasure Coding を使用したものであっても書き込みと同時に符号化は行われないなど，本研究が取り組む内容とは異なっている．また，本稿で示す実装については，通信のほぼすべてが RDMA を活用したものとなっており，メモリ帯域の消費やオーバヘッドを極限まで抑える取り組みも他と異なるものである．

3. Cluster-wide RAID

3.1 概要

Cluster-wide RAID は，元々ノード内における記憶装置の冗長性を確保するために開発された RAID の考え方を，

ネットワーク上に展開してストレージノードレベルの冗長性を確保するものである，元々 RAID はノード内のディスク故障やセクタ不良に備えるために開発されたものであることから，ノード故障についてはそもそも対象としていない．このため，ノードレベルの障害に対しては，ネットワークを介してデータ交換を行い，複製や冗長符号の生成を行う必要がある．典型的なシステムの構成においては，ノード内におけるディスク間の接続に比べると，Ethernet を始めとするネットワークの帯域やレイテンシといった性能は優れないことが多い．そのため，RAID に似たデータ構造を直接分散ストレージシステム上に構成しても，十分な性能を得るのは困難であった．しかしながら，筆者らは InfiniBand の RDMA 機構を最大限活用し，最適なデータ構造を利用することで，高い性能を確保しており，これについては既に数件の発表を行っている [3][9].

3.2 全体の構成

Cluster-wide RAID の基本的な構成を図 1 に示す．各ストレージノードは，RAID と同じように配置されたデータを保存する．例えば，図のケースにおいては，ストライプ化された元のデータと，パリティをそれぞれを保持する．各ストレージノードとクライアントはネットワークで接続されているが，通常の RAID と異なる点は，ストレージノード間にもデータを転送するパスが存在する点である．ディスクはデータを保存/アクセスするだけであるが，Cluster-wide RAID においては独立した 1 つの計算機がストレージになるため，演算やデータの送信/交換が可能となり，この点が本質的な差異をもたらしている．分散型のアプローチにおいては，このストレージノード間において積極的に演算やデータ交換を行うことにより，パリティ生成などのトラフィック増加や演算を伴う処理について，集中を防いだ．

一方で，本稿において提案する集中型のコントローラを用いると，従来の RAID と近い形となるが，依然としてそれぞれは計算機であるため様々な処理を行う可能性は残されている．将来的には，集中型と分散型のアプローチを併用する形で，最適なストレージネットワークを構成することを目指している．

4. コントローラの提案

4.1 分散型の手法と集中型コントローラ

これまでの研究においては，分散型の手法で高速な冗長符号の生成を行っており，その概要は図 2 の右図に見られる通りである．本稿を含め，最適化の対象は Cluster-wide RAID の書き込みフェーズにおけるオーバヘッドの最小化である．分散型のアプローチにおいては，データをバケツリレーのように回しながら xor 演算を行い，必要な冗長符号を生成することにより，トラフィックや演算負荷の局所

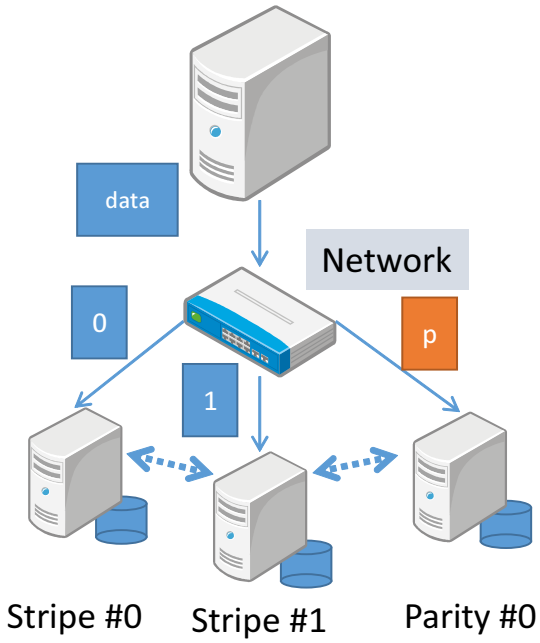

図 1 Cluster-wide RAID-4 の構成例

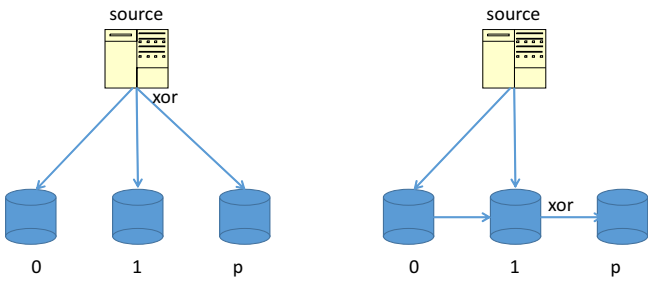

図 2 分散型の Cluster-wide RAID 最適化 (右) [3] より

化を回避した．結果として，冗長符号の生成の有無にかかわらず，高い性能を確保することが出来た．

一方で，本稿が提案する集中型のアプローチにおいては，ネットワーク上に Cluster-wide RAID に関する処理を行うコントローラを導入する，このコントローラは他のストレージノードやクライアントを比較して大きなネットワーク帯域と豊富な演算能力を持ち，データ量の増大を伴う処理をまとめて引き受ける．この構成を図 3 に示している．現時点においては，このコントローラは通常の計算機である．将来的には，プログラマブルで処理能力を持つネットワークスイッチなどが登場した際に，このような手法をそのまま導入することが可能となり，柔軟かつネットワーク指向のストレージシステムを構成することが出来る．

4.2 計算機を用いたコントローラの構成

本研究が現在ターゲットとしている Cluster-wide RAID の書き込みフェーズについては，パリティにより最終的なデータ量が増大し，その転送と記録のためのオーバヘッド

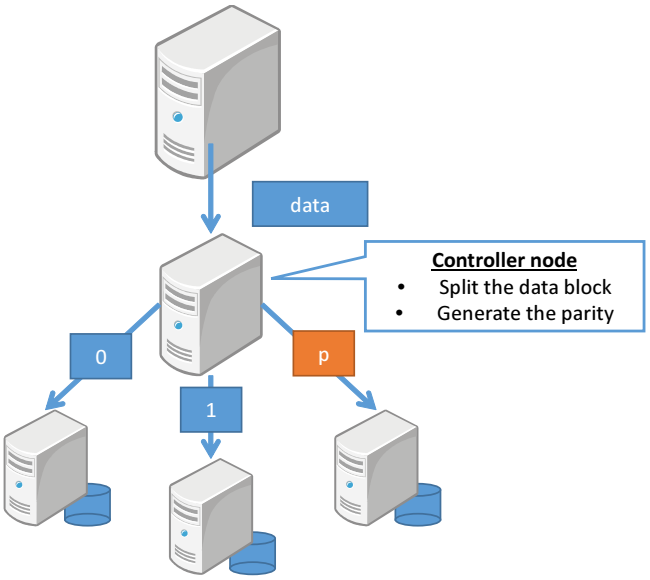

図 3　Cluster-wide RAID 向けのコントローラ

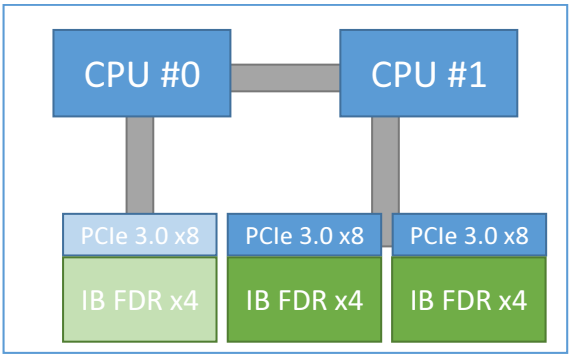

図 4　コントローラノードの構成（色の薄い部分は現時点で装着していない）

CPU	Intel(R) Xeon(R) CPU E5-2665 x2
RAM	64GB
InfiniBand HCA	Mellanox MT27500 4x FDR (56Gbps) (コントローラノードには 2 つ搭載)

図 5　評価環境

が生じる点が問題であった．書き込み元のクライアントからナイーブに元データとパリティを送信すると，パリティによりデータ量が増大した分だけスループットが低下する．したがって，分散型のアプローチにおいては，ストレージノード間でデータを交換しつつパリティを生成し，クライアントは元データのみをストレージノード群に送信することでトラフィックの増大を防いでいた．一方で，本稿において提案するネットワークストレージコントローラは，パリティの生成とストライプの配布をすべて引き受けるものである，これは，RAID コントローラにも似たものであるが，ネットワーク上に展開している点で異なる．

現時点ではこのような処理を行う専用のコントローラは存在しないため，通常の計算機を用いて構成した．用いた計算機の構成を図 4 に示す．通常の計算機に，InfiniBand FDR HCA を 2 枚装着し，他のクライアントやストレージノードと比べて多くのバンド幅を確保した．これらのネットワークは全二重であるため，双方向にすべての帯域を使うことが可能である．

実際にデータの書き込みを行う際には，書込元クライアントとストレージノード群の間にコントローラを配置して使用する．コントローラは書込元クライアントから受け取ったデータをストライプに分割するとともに，必要なパリティを生成し，それらをストレージノード群に送信する．

4.3　実装

実装に関しては，これまでと同じく，転送のほぼすべてを Verbs API を用いて記述し，RDMA を最大限活用している．実装の概要は図 3 に示すものと同じで，これは RAID-4 (2+1) を構成する場合の例である．分散型の実装でも用いたパイプライン構造を応用しており，DMA に使用するメモリ領域はリングバッファを構成している．書込元ホストとコントローラの受信側については，単に元のデータをやりとりする．コントローラは受信したデータを分割し，ストライプとしてストレージノードに送信すると共に，それらの xor 値を求め，パリティ用のストレージノードに転送する．これらの操作のうち，ストライプを送信する部分に関してはコピーを行わずにダイレクトに送信する事も可能であるが，本実装では用いている独自ライブラリの仕様上，一旦コピーを行っている．

4.4　性能の見積もり

トラフィックの増大を伴う処理を，豊富なバンド幅を持つノードに移動させることにより，ボトルネックを解消するのが本提案の基本的な発想である．この節では，各 RAID のケースについて，コントローラに求められるバンド幅について見積を行う．元の書き込むデータ量を m とする場合，RAID-4 (n+1) においては，データ量は $m(1 + 1/n)$ に増大する．コントローラは元のデータを受け取った上でパリティを付加してストレージノード群にデータを送信するので，書込元ノードが持つバンド幅と同等の受信帯域と，$(1 + 1/n)$ 倍の送信帯域があれば，性能低下なく書き込みを行える．複数の書き込み元がある場合も同様であり，書込元ホストの合計送信帯域に対して同じ見積をすることで，コントローラに必要な帯域を求めることが出来る．尚，本稿における評価では，コントローラの帯域を受信/送信ともに書き込み元の 2 倍とした．構成の詳細については評価の章で述べる．

5.　性能評価

5.1　評価環境

評価に用いた環境は，図 5 に示す通りである．いずれのノードも InfiniBand FDR HCA を搭載しており，単一

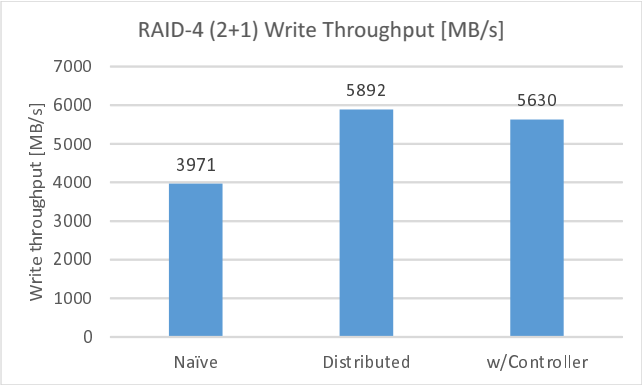

図 6　128KB ブロックサイズにおける RAID-4 (2+1) の性能比較

の接続では全二重 6GB/s ほどの実測スループットである．コントローラ以外のノードにおいては，1 つの HCA を搭載している．コントローラとして用いるノードには，2 枚の HCA を装着した．使用しているのはマルチソケットのサーバであり，CPU 間の通信を避けるため，同一の CPU 側につながるようスロットを選んで配置した．尚，いずれの評価も記憶装置の性能を大幅に上回っており，加えて，今回の評価で注目しているのはネットワークバンド幅の制約による性能低下のため，最終的なバッファからディスクへの書き込みは省略している．測定結果については，5 回の平均値を用いている．

5.2　RAID-4 2+1 の書き込み性能

図 6 は，RAID-4 (2+1) の各ケースにおける書き込みスループットを示したものである．このケースでは，2 台のストレージノードにストライプを記録し，残りの 1 台がそれらのパリティを保存する．評価にあたっては，書き込み元のクライアンノードがストライプの分割およびパリティの生成を行うナイーブな方法，筆者らがこれまでに提案した分散型の最適化による性能，そして最後に本稿で提案するコントローラを用いた場合の性能について比較した．

5.3　考察

ネットワーク性能は約 6GB/s が最大値であり，Naive に関してはデータの増加分だけ性能が低下し，約 4GB/s となっている．Distributed はこれまでに提案した分散型の最適化手法を適用した場合の結果であり，ほぼネットワーク性能に近い値となっている．最後の w/Controller が本稿において提案するコントローラノードを導入した場合の性能である．実装の最適化にまだ余地があるため，Distributed と比較すると若干の低下が見られるが，Naive と比較すると約 42%と大幅に性能が向上しており，提案手法の有効性が分かる．

6.　まとめと今後の課題

6.1　まとめ

本稿では，分散型のストレージシステムにおいてノードレベルの冗長性を提供する Cluster-wide RAID を対象として，その性能低下を抑える書き込み手法を提案した．筆者らのこれまでの提案は分散型のアプローチを用い，ストレージノード群の内部でデータ交換を行うことにより最適化していたが，本稿においては豊富なバンド幅を持つ集中型のコントローラを導入することで性能低下の解消を図った．実際の評価においても，ナイーブな実装よりも大幅に性能が向上し，これまでに提案した手法と遜色ない性能を達成することができた．

6.2　今後の課題

本稿にて提案した手法と，筆者らがこれまでに提案した手法は，併用することも可能である．今後はそれぞれのメリット・デメリットを詳細に分析した上でハイブリッドな手法についても検討したい．さらに，大規模なシステムにおいてはこのようなコントローラは 1 つではなく，多数存在することになるため，その協調動作に関してもさらなる研究が必要である．加えて，今回の実装においてはコントローラ上において完全なゼロコピーを実現できていないため，データ構造の見直しなども必要である．また，ストレージノード群においてコントローラを配置したことにより，I/O delegation [10] といった手法を併せ持つことも可能になっており，この検討はさらなる性能向上の点からも有益であると考えている．

謝辞　本研究の一部は，JSPS 科研費 (特別研究員奨励費)26・1967，JST CREST「ポストペタスケールデータインテンシブサイエンスのためのシステムソフトウェア」，JST CREST「EBD：次世代の年ヨッタバイト処理に向けたエクストリームビッグデータの基盤技術」による．

参考文献

[1] Chervenak, A. L., Foster, I. T., Kesselman, C., Salisbury, C. and Tuecke, S.: The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets, *Journal of Network and Computer Applications*, Vol. 23, pp. 187–200 (1999).
[2] Patterson, D. A., Gibson, G. and Katz, R. H.: A Case for Redundant Arrays of Inexpensive Disks (RAID), *SIGMOD Rec.*, Vol. 17, No. 3, pp. 109–116 (1988).
[3] 大辻弘貴，建部修見：分散ストレージシステムに対する低オーバヘッド冗長化書き込み手法の提案と評価，情報処理学会研究報告ハイパフォーマンスコンピューティング HPC142，pp. 1–6 (2013).
[4] Tatebe, O., Hiraga, K. and Sod, N.: New Generation Computing, Ohmsha, Ltd. and Springer, *Gfarm Grid*

File System, Vol. 28, No. 3, pp. 257–275 (2010).
[5] Braam, P. J.: Lustre, *http://www.lustre.org/*.
[6] Hadoop: *http://hadoop.apache.org/*.
[7] Shvachko, K., Kuang, H., Radia, S. and Chansler, R.: The Hadoop Distributed File System, *Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, MSST '10, Washington, DC, USA, IEEE Computer Society, pp. 1–10 (2010).
[8] Fan, B., Tantisiriroj, W., Xiao, L. and Gibson, G.: DiskReduce: RAID for Data-intensive Scalable Computing, *Proceedings of the 4th Annual Workshop on Petascale Data Storage*, PDSW '09, New York, NY, USA, ACM, pp. 6–10 (2009).
[9] 大辻弘貴，建部修見：Cluster-wide RAID の実装と評価，情報処理学会研究報告ハイパフォーマンスコンピューティング HPC145，pp. 1–5 (2014).
[10] Nisar, A., keng Liao, W. and Choudhary, A.: Delegation-Based I/O Mechanism for High Performance Computing Systems, *Parallel and Distributed Systems, IEEE Transactions on*, Vol. 23, No. 2, pp. 271–279 (2012).