様式Ｃ－１９

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成 24 年　4 月　1 日現在

機関番号： 34419
研究種目：若手研究（B）
研究期間：2009～2011
課題番号：21700063
研究課題名（和文）　負荷分散に配慮した透過的な計算機間大規模高速並列入出力に関する研究
研究課題名（英文）　Study of a High Performance Seamless Parallel I/O System between Parallel Computers with Load Balancing Awareness
研究代表者
辻田　祐一（TSUJITA　YUICHI）
近畿大学・工学部・准教授
研究者番号：70360435

研究成果の概要（和文）：
MPI における並列入出力である MPI-IO に集団型並列入出力がある。この中で用いられる最適化手法に対し、Pthreads によるマルチスレッド方式および非同期 I/O API 方式の 2 種類を提案し、オリジナル実装よりも高性能な I/O が可能であることを確認した。また各プロセスの処理時間の不均一さが、プロセス全体で同期を取る際に遅いプロセスに全体の処理が同期してしまう問題があり、全体の性能を低下させる問題があった。そのため、本研究では、マルチバッファリングによる遅延の影響を受けにくい実装を行い、その有効性を確認した。

研究成果の概要（英文）：
Parallel I/O interface in the MPI standard named MPI-IO provides collective parallel I/O. We have developed two pipelined processing implementations based on an existing original implementation by using (1) multithreaded technique with the help of Pthreads and (2) asynchronous I/O APIs, and both implementations have outperformed the original implementation. Furthermore, we have addressed to utilize a multiple buffering mechanism to minimize idle times of each process, and it has outperformed the original one.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
2009 年度	2,700,000	810,000	3,510,000
2010 年度	500,000	150,000	650,000
2011 年度	500,000	150,000	650,000
年度			
年度			
総　計	3,700,000	1,110,000	4,810,000

研究分野：総合領域
科研費の分科・細目：情報学、計算機システム・ネットワーク
キーワード：ネットワークコンピューティング、並列入出力、PC クラスタ

１．研究開始当初の背景

並列計算機を用いたシミュレーションは、年々規模が増大しており、それに伴いデータ量も増加している。そのため、データの入出力がボトルネックになるため、近年は並列ファイルシステムを配置させ、並列 I/O インタフェースによる高速入出力が行われている。並列計算で標準的に用いられている通信インタフェースである Message Passing Interface(MPI)でも、並列 I/O インタフェースである MPI-IO が制定されており、代表的

な実装として ROMIO がある。

特に並列プログラムにおいては集団型並列 I/O が広く用いられているが、計算アプリケーションが扱うデータが多次元データ等の場合、ファイルアクセスでは不連続なアクセスパターンになる。このような場合、ROMIO では、図 1 に示すような Two-Phase I/O と呼ばれる最適化手法が用いられている。

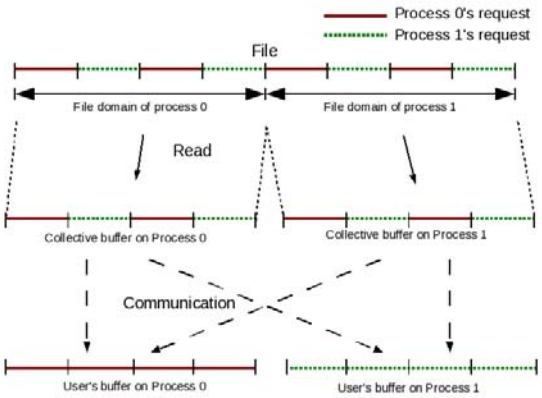


図 1：Two-Phase I/O の処理フロー

この手法では、各プロセスでファイルアクセス領域が連続になるように均等に分割し、Collective Buffer（以下、CB）と呼ばれる中間バッファ単位にファイルアクセスを複数回（以下、1 回分のアクセスをサイクルと呼ぶ。）行い、ファイルアクセスの毎に、必要とするデータをプロセス間通信によりユーザ・バッファに格納させる。これにより不連続パターンによる膨大なファイルアクセス回数を大幅に減らすことができ、性能が向上する。しかしながら、通信と I/O が逐次的に交互に繰り返し、かつ集団型 I/O のために、プロセス間の同期も入り、各々のプロセスの負荷バランスが悪くなるにつれて、益々遅延が発生し、思うように性能向上が望めない問題があった。

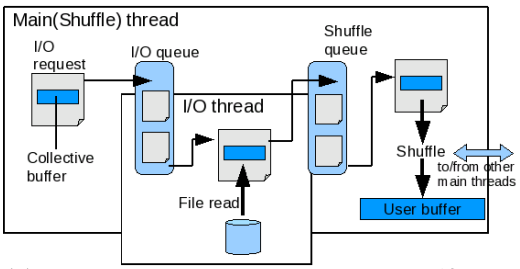
2．研究の目的

1 で述べたような問題を解決し、高性能な並列 I/O が行えるように高速化実装を検討する。試験実装を行うことにより、将来の並列 I/O の高速化に向けた実装の方向性を見出すことも本研究の主目的である。

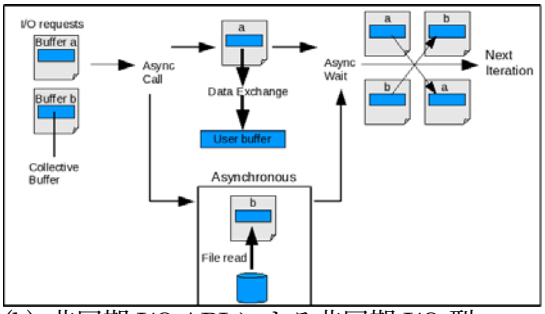
3．研究の方法

PC クラスタ等で広く利用可能で、容易に実装が利用できることを重視し、既存の Two-Phase I/O の実装に改良実装を行うことにした。Two-Phase I/O は ROMIO 内部の共通 I/O レイヤである ADIO ライブラリレイヤ内の MPI-IO インタフェースにリンクする共通インタフェースで実装されている。そのため、この改良実装は ROMIO がサポートする様々なファイルシステムに対し利用可能になる汎用的な利点もある。

本研究では、図 2 に示すパイプライン型 Two-Phase I/O の実装を行った。ここでは (1)Pthreads によるマルチスレッド処理による図 2(a)の実装と(2)非同期 I/O API による図 2(b)の実装の 2 種類を検討および開発した。



(a) Pthreads によるマルチスレッド型



(b) 非同期 I/O API による非同期 I/O 型

図 2：パイプライン型 Two-Phase I/O 実装の動作イメージ （(a)Pthreads によるマルチスレッド型および(b)非同期 I/O API による非同期 I/O 型）

前者の実装では、MPI による通信を行うメインスレッド（図中の Main(Shuffle) thread）と I/O 処理を行う I/O スレッド（図中の I/O thread）の 2 つを配置し、I/O キュー（I/O queue）とデータ交換キュー（図中の Shuffle queue）を両スレッド間で共有する。Main thread が発行した I/O 要求は I/O queue にキューの最大の深さまで投入される。一方、I/O thread は、このキューに要求があると、それを順に取り出し、要求に基づいた I/O 処理を行い、終了のたびに Shuffle queue に要求を投入する。Main thread はこのキューから順に要求を取り出し、プロセス間のデータ交換を行う。これを全ての要求が完了するまで繰り返す。

一方、後者の実装では、並列ファイルシステムが有する非同期 I/O API を用いてデータ通信と I/O 処理のオーバーラップを実現している。実装上の制約から、I/O queue や Shuffle queue は配置されず、前者の実装のように複数の I/O 要求やデータ交換要求を溜

め込みながら処理をすることはできない。ただし、ファイルシステムが提供する非同期 I/O API が計算処理との高いオーバーラップを実現できれば、大変有効な手法となる。

４．研究成果

1 年目の成果としては、並列 I/O の性能向上に向け、I/O 処理のパターンをトレースすることで得られる情報の分析を行うと共に、負荷バランスを考慮した高速並列 I/O に向けた検討を行った。その結果、3 で述べている Pthreads 及び非同期 I/O API を用いた 2 つの方式を用いて実現することを検討した。

次に 2 年目の成果としては、1 年目の検討をベースに試験実装を行い、PC クラスタにおいて評価試験を行った。ここでは、ユーザプロセスが発行する I/O 要求を同時に複数バッファメモリに保留し、特に非同期的にどの程度効率良く機能するかも評価した。その性能評価の一例を図 3 に示す。

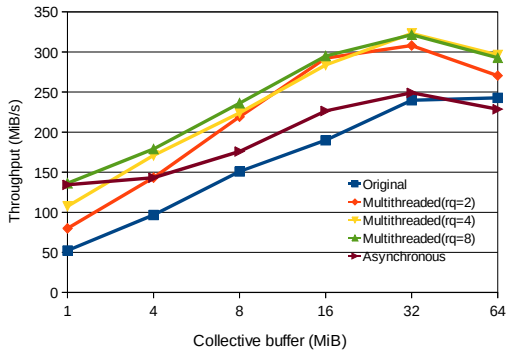
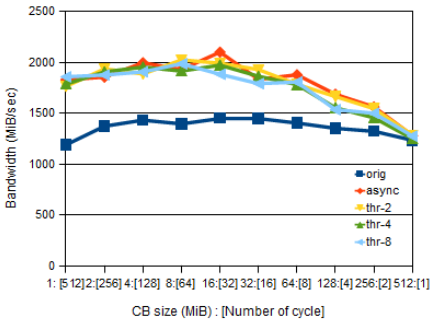


図 3：8 個のプロセスによる PVFS2 への集団型並列読み込みの性能

この図は、8 個の MPI プロセスを起動し、メタデータサーバ 1 台、データサーバ 8 台で構成された PVFS2 ファイルシステムに対して MPI-IO による集団型並列読み込み性能を示している。測定では CB の大きさを変えながら計測しており、ある程度 CB を大きくしておけば、データ領域すべてを網羅するようなユーザ・バッファを確保しなくても性能が向上する可能性があることが分かる。この図で、特にマルチスレッド型の性能上の優位性が見て取れる。一方、非同期版はマルチスレッド版ほど性能向上が出来なかったが、それなりの性能上のメリットがあることは確認された。この評価を通して、低レイヤにある並列ファイルシステムの有無に関わらず、実装の持つ大きな優位性を発揮できることを確認した。

最後に 3 年目の成果としては、それまでに判明した問題点を整理し、完成度を高めた実装を行った。また、並列ファイルシステムとして小さい PC クラスタでは PVFS2 を対象としていたが、より大規模な並列計算機での評価も重要であるため、東京大学情報基盤センターにある T2K オープンスーパーコンピュータを利用し、並列ファイルシステムの一つである Lustre への並列 I/O 性能を評価した。その結果の一例を図 4 に示す。

(a) 8 プロセス（1 プロセス/ノード）



(b) 64 プロセス（1 プロセス/ノード）

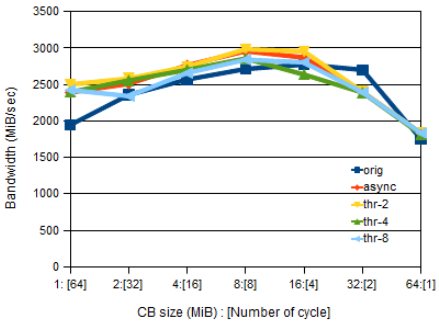


図 4：(a) 8 個のプロセス及び (b) 64 個のプロセスによる Lustre に対する集団型並列読み込み性能（横軸の括弧内は Two-Phase I/O のサイクル数）

この図の 8 プロセスの場合では、提案手法がオリジナル実装を大きく上回る結果となった。より少ない CB で大幅な性能向上が見込めることが分かった。一方 64 プロセスの場合、前者ほどの差は開かなかったが、CB の大きさが小さめに設定した時は、依然として提案手法の方が高い性能を示していた。ただし、使用した Lustre の物理的なスループット限界がおおよそ 3 GB/s のため、性能向上が頭打ちになり、オリジナル実装との差があまり出ていなかったと思われる。また CB が 32MiB の時は、オリジナル実装が高い性能を示していた。この場合、多数のプロセスによる集団型 I/O におけるプロセス間の同期のコストが大きくなっている。提案手法ではオーバーラップ効果が小さくなり、さらにパイプライン処理のオーバーヘッドのコストも目立つようになるのに対し、オリジナル実装に

おいては、余分な処理をすることなく、Lustre の先読み機能が有効に働くようになったことで、オリジナル実装の方が高い性能を示していたものと考えられる。

以上の結果、本研究での提案手法の有効性が、ある程度の規模の PC クラスタにおいて検証できた。また同時に保有する I/O 要求数を複数にすることで、集団型並列 I/O で起こりがちな遅延の蓄積を緩和することも確認できた。

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計 5 件）

① Yuichi Tsujita, Hidetaka Muguruma, Kazumi Yoshinaga, Atsushi Hori, Mitaro Namiki, and Yutaka Ishikawa, “Improving Collective I/O Performance Using Pipelined Two-Phase I/O,” Proceedings of SCS Spring Simulation Conference 2012 (HPC2012), 査読有, March 26-29, 2012, Orlando, Florida, USA (CD-ROM)
② Michael Kuhn, Julian Kunkel, Yuichi Tsujita, Hidetaka Muguruma, Thomas Ludwig , “Optimizations for Two-Phase Collective I/O,” Book of Abstracts of ParCo2011, 査読有, Ghent, Belgium, August 30-September 2, 2011, pp. 147
③ 六車　英峰, 辻田　祐一, “ パイプライン処理による Two-Phase I/O の高速化 ” 第 12 回 IEEE 広島支部学生シンポジウム（12th HISS）論文集, 査読有,（2010 年 11 月 6-7 日、島根大学（島根県松江市））(CD-ROM)
④ Yuichi Tsujita, Julian Kunkel, Stephan Krempel, and Thomas Ludwig, “Tracing Performance of MPI-I/O with PVFS2: A Case Study of Optimization,” Parallel Computing: From Multicores and GPU’s to Petascale (Post Proceedings of ParCo’09), IOS Press, 査読有, Vol. 19, 2010, pp. 379-386
⑤ Julian M. Kunkel, Yuichi Tsujita, Olga Mordvinova, and Thomas Ludwig, “Tracing Internal Communications in MPI and MPI-I/O,” Proceedings of PDCAT 2009, 査読有, Hiroshima, Japan, December 8-11, 2009, IEEE CS Press, pp.280-286

〔学会発表〕（計 7 件）

① Hidetaka Muguruma, Yuichi Tsujita, “Study of A Pipelined Processing for High Throughput Collective MPI-I/O,” 東大 T2K 若手利用者推薦制度（2011 年度前期）成果報告会（第 11 回 ASE 研究会内で併設実施）（2012 年 3 月 16 日, 東京大学情報基盤センター, 東京都文京区）
② 六車　英峰, 吉永　一美, 辻田　祐一, 堀　敦史, 並木　美太郎, 石川　裕, “ パイプライン型 Two-Phase I/O の Lustre における性能評価 ”（口頭発表）, HOKKE 2011（2011 年 11 月 28 日-29 日, 北海道大学, 北海道札幌市）情報処理学会研究報告 Vol. 2011-HPC-132, No. 35（オンラインドキュメント）
③ 六車　英峰, 辻田　祐一, 堀　敦史, 並木　美太郎, “ Two-Phase I/O の高速化に関する一検討 ”（口頭発表）, SWoPP 2011（2011 年 7 月 27 日-29 日, 鹿児島県民交流センター, 鹿児島県鹿児島市）情報処理学会研究報告 Vol. 2011-HPC-130, No. 132（オンラインドキュメント）
④ 六車　英峰, 辻田　祐一, “ マルチスレッディングによる高速な集団型並列入出力の実装 ”（ポスター発表）, 次世代スーパーコンピューティングシンポジウム 2010 および第 1 回戦略的プログラム 5 分野合同ワークショップ（2011 年 1 月 17 日, ニチイ学館　神戸ポートアイランドセンター　大会議室（兵庫県神戸市））
⑤ 六車　英峰, 辻田　祐一, “ パイプライン処理による Two-Phase I/O の高速化 ” (ポスター発表), 第 12 回 IEEE 広島支部学生シンポジウム（12th HISS）（2010 年 11 月 6-7 日、島根大学（島根県松江市））査読あり
⑥ 六車　英峰, 辻田　祐一, “ マルチスレッドによる Two-Phase I/O の高速化の検討 ”（口頭発表）, 情報処理学会第 127 回ハイパフォーマンスコンピューティング研究発表会（2010 年 10 月 13 日、理化学研究所・和光本所（埼玉県和光市））, 情報処理学会研究報告 Vol. 2010-HPC-127, No. 6（オンラインドキュメント）
⑦ Hidetaka Muguruma, Yuichi Tsujita, Julian M. Kunkel, and Thomas Ludwig, “Tracing Temperature of Hard Disk Drives for Monitoring Status of PVFS2,” International Supercomputing Conference 2010 (Hamburg, Germany, May 31-June 3, 2010) 査読有

〔その他〕
ホームページ等
http://hpclab.hiro.kindai.ac.jp/

６．研究組織
(1)研究代表者
辻田　祐一（TSUJITA　YUICHI）
近畿大学・工学部・准教授
研究者番号：70360435