情報処理学会研究報告
IPSJ SIG Technical Report

Vol.2011-ARC-197 No.16
Vol.2011-HPC-132 No.16
2011/11/28

Non-blocking RPC を用いた 遠隔ファイルアクセスの実装と性能評価

大辻弘貴[†1,†2] 建部修見[†1,†2]

ネットワークを経由したファイルアクセスの最適化について，我々はこれまでに同期型 RPC を用いた場合の性能評価および最適化手法について取り組んできた．また，非同期型 (Non-blocking)RPC を用いた場合の実装や性能の見込みについても発表を行ってきた．本論文においては，非同期型 RPC を用いた遠隔ファイルアクセスについて，その実装と性能評価について述べる．シーケンシャルアクセスやストライドアクセスなどのパターンにおいて，実測した性能測定結果について示すと共に，さらなる最適化のための手法について考察する．

1. 序　　論

複数の拠点間で取り扱うデータが急増すると共に，ネットワークを経由して大容量データを共有する機会が増えている．特に，データインテンシブコンピューティングや e-サイエンスの発展は，広域分散環境の活用を後押ししている．このような環境の中では広域分散ファイルシステムによってデータの共有システムを構築する事があるが，実際の広域環境においてはネットワークのバンド幅はもとより，回線遅延による性能低下などの問題がある．このような問題に対応するため，1) に示すようなファイル複製によるデータアクセスの手法がある．この方法はファイル単位でのデータ利用であれば問題無いが，ファイルの一部分のみを必要とする場合やシンクライアントなどのローカルストレージの乏しい端末から利用する場合などにおいて，効率性や実現性の面から問題が生じる可能性がある．このようなケースでは，アプリケーションの I/O 要求をネットワーク越しに行う遠隔ファイルアクセスを行うことになるが，ファイル複製の際に行われるバルク転送と異なり，オペレーションを逐一ネットワークを介して伝送するので，特に回線遅延の影響を受けやすい．この問題に対応するためのパラメータなどが用意されたソフトウェアはいくつか存在するが，管理者が手動で設定しなければならない，あるいは固定値であるため利用形態や環境の変化に対応できないといった問題がある．本論文では，既存システムが抱えるこれらの問題について，評価を踏まえた上で，ネットワーク環境や利用形態に応じてアダプティブな最適化を行うための手法について考察する．

†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 独立行政法人科学技術振興機構 CREST
JST CREST

2. 関 連 研 究

序論でも述べたように，本論文においてはデータ共有の方法として遠隔ファイルアクセスを用いる場合を対象としている．遠隔ファイルアクセスの方式も更に分類することができる．これは，RPC によって逐一リクエストを発行する方式と http などのようにファイル全体を転送する方式である．それぞれ例を挙げると，前者には Gfarm[2] ファイルシステムや NFS[3]，Lustre[4] などがある．後者には AFS[5] やその後継である Coda[6]，FTP をグリッド環境向けに拡張した GridFTP[7] が挙げられる．本論文は前者の RPC を用いる場合を対象としている．

上記に挙げたシステムのうちいくつかは性能を向上するための仕組みが備わっている．例えば，Lustre は，回線遅延の大きいネットワークに対応するため，RPC in flight というパラメータがあり，応答無しに発行できる RPC (pending request) の数を設定する事が出来る．ただし，この値は固定値であり，管理者が手動で設定し，動的に変更されるようなことは無い．

3. 同期型 RPC・非同期型 RPC の性能比較

前章で述べたように，遠隔ファイルアクセスに用いられる RPC を細かく区分すると，同期型と非同期型の RPC がある．同期型 (blocking)RPC では，リクエストはクライアントから 1 つずつ発行され，サーバから結果が返されるまで他の動作は行えない．一方の非同期型は処理の状況にかかわらずリクエストを連続的に発行することができる方式である．本論文では，後者の方式を取り入れた上で，遠隔ファイルアクセスの最適化を行った．

前提条件として，クライアント上で動作するアプリケーションは同期的に I/O リクエストを行うものとする．すなわち，アプリケーションが予め必要なデータをまとめてリクエストするような事は無い．このような条件に元で，遠隔ファイルアクセスのシステムが非同期に RPC を発行する．本章では各 RPC の方式による性能について示す

Vol.2011-ARC-197 No.16
Vol.2011-HPC-132 No.16
2011/11/28

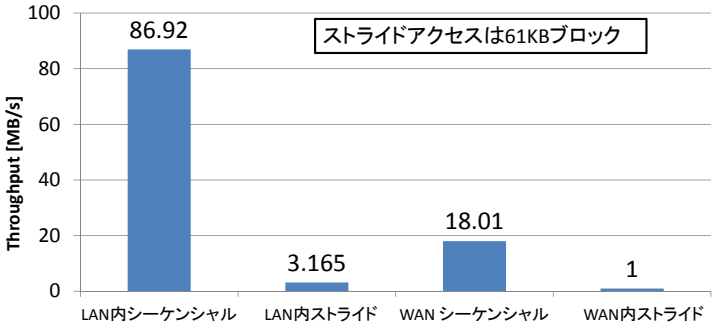

図 1 既存システム[2] における性能
LAN のネットワーク遅延は 50 μ s，WAN は 25ms
Fig. 1 Performance of existing method.
Network delay of LAN is 50us and WAN is 25ms.

3.1 性能評価の方法

性能評価に際しては，通信レベルでの最適化を行うために，既存の分散ファイルシステムなどは用いず遠隔ファイルアクセスの評価のみに注目したプログラムを作成した．**図 2** は，評価に用いたプログラムの構成である．コンピュータはクライアントとサーバの 2 台から成り，クライアントアプリケーションのアクセスは，FUSE[8](Filesystem in Userspace) を介して遠隔ファイルアクセスのシステムが処理する．この方法では，OS のファイルアクセスの API を利用できるため，ファイルシステム上で動作するアプリケーションであればそのまま動かすことが可能である．FUSE を介して発行されたリクエストは，処理を行った上で，ネットワークを経由してサーバに送信される．サーバはその応答を返し，アプリケーションは FUSE を介してその結果を受け取る．本論文で述べる手法については，大部分が図中の Remote File Access の部分において実装される．

評価はクライアントとサーバの 2 台のコンピュータを用いて行い，それぞれが Gigabit Ethernet で接続されている．サーバ側のストレージには，7200 回転の HDD2 台を接続してストライピングを行い，ネットワーク以上のバンド幅を確保できることを確認している．

3.2 RPC による遠隔ファイルアクセス

図 3 に本研究が対象とする非同期型 RPC によるファイルアクセスの手順を示す．クライアントアプリケーションが発行した I/O 要求はファイルシステムのクライアントに送られ，

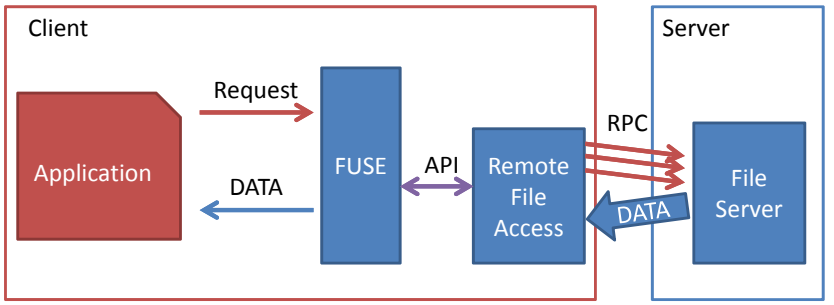

図 2 評価システムの構成
Fig. 2

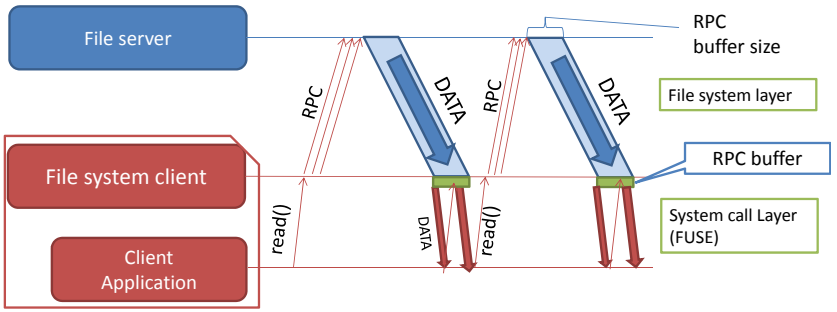

図 3 非同期 RPC による遠隔ファイルアクセス
Fig. 3

RPC が発行される．ここでは非同期型の RPC を用いているため，サーバからの応答を待たずに複数の RPC を発行することができる．この図中においては例として 3 個の RPC が発行されている．また，サーバから返された応答データは，一旦ファイルシステムクライアントの RPC バッファと呼ぶ領域に保存される．クライアントアプリケーションは FUSE を介して，このバッファ領域にアクセスし，要求していたデータを得る．この RPC バッファサイズの最適化については，筆者らがこれまで進めてきた研究の一部である．

3.3 回線遅延の影響

回線遅延が増大すると，性能が低下する傾向にある．これは，クライアントとサーバの間で通信に時間がかかるようになり，相対的にデータを転送出来る時間が減少するからであ

Vol.2011-ARC-197 No.16
Vol.2011-HPC-132 No.16
2011/11/28

る．この影響は同期型 RPC を用いていて，一度に転送するデータ量が少ない場合に顕著になる．非同期型 RPC を用いて，適切に先行してリクエストを発行することにより，このような性能低下を抑えることが出来る．

図 4 は，同期型 RPC と非同期型 RPC のそれぞれによるファイルアクセスについて性能を比較したものである．横軸はサーバ・クライアント間の回線遅延，縦軸は測定されたスループットを示す．アクセスパターンはシーケンシャルアクセスである．同期型 RPC については RPC バッファサイズ（一度の RPC で転送するデータ量）を 64MB とし，同期型 RPC については同 RPC バッファサイズを 1.5MB，Pending request の数（応答を待たずに要求するリクエスト数）を 40 とした．非同期型 RPC の場合について，Pending request と RPC バッファサイズの積は 60MB となり，バッファの総量ではほぼ同等となる．非同期型 RPC の場合に先行して発行する要求は，クライアントアプリケーションが要求した位置の直後としている．

この図から分かることは，非同期型 RPC においては，0〜25ms の間では性能低下を抑えられていることである．また，スループット自体も高い．しかしながら，50ms の遅延では性能が逆転している．この原因はまだ完全に明らかには出来ていないが，連続したデータ転送を行っている場合，サーバ・クライアント間で TCP を用いた場合の通信遅延が 2 倍から 3 倍になることがあった．このような遅延の揺らぎによって非同期リクエストの送受信に中断があった可能性がある．

4. 提案手法および評価

筆者らは，同期型 RPC および非同期型 RPC を用いた遠隔ファイルアクセスの最適化の研究の中で，アクセスパターンの認識方法を既に提案している．同期型 RPC に対しては，回線遅延と帯域を考慮した上で RPC バッファサイズを決定するためのアクセスパターン検知手法を，非同期型 RPC についてはアクセスパターンの規則性を検知する手法について提案した．

本論文においては，筆者らのこれまでの研究で示したストライドアクセスパターン検知手法の実装について述べる．

ストライドアクセスとは，あるアクセスとシークの繰り返しから構成されるアクセスパターンである．同期型 RPC を利用した場合には，アクセスされる領域を一つずつ転送するか，シークされる領域も含めてまとめて転送するかのどちらかになる．前者の方法では，領域が小さい場合に回線遅延の影響を大きく受ける．また，後者の方法ではアクセス量よりもシーク量の方が多かった場合に，無駄な転送が多く発生し，実効スループットが大きく落ち込む．

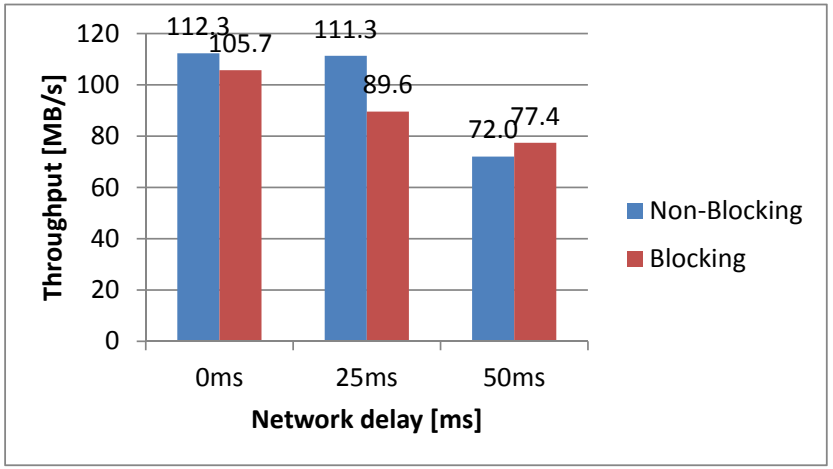

図 4 Blocking RPC と Non Blocking RPC の性能比較
Fig. 4

このような場合，非同期型の RPC を使い，検知したアクセスパターンに基づいて先回りして転送 (read ahead) を行う事が出来れば高い性能が達成できると考えられる．**図 5** は，検知したパターンに基づいて転送する様子を表した図である．読み込まれると（予測される場所）を予め非同期 RPC で複数指定してリクエストを発行する．この方法を用いた場合の性能向上を検証するために，以下に示す評価を行った．

図 6 は，3MB の読み込みと 6MB のシークを繰り返すストライドアクセスにおいて，同期型 RPC で最適な RPC バッファサイズを使用した場合と，非同期型 RPC を用いて先読みを行った場合とを比較した評価結果である．縦軸はスループット [MB/s] であり，横軸はサーバとクライアント間の回線遅延 [ms] である．同期型 RPC のデータについては，既発表の論文から引用している．非同期型 RPC の先読みについては，アクセスパターン検知手法の実装が完了していなかったため，予め先読み位置を指定した上で実行している．このような一定ストライドアクセスパターンの検知が可能であることは，既に筆者らのこれまでの研究で示している．先読み位置は，ファイル中の位置で 300MB 先までとしている．すなわち，3MB の読み込みブロック約 35 個分である．

Vol.2011-ARC-197 No.16
Vol.2011-HPC-132 No.16
2011/11/28

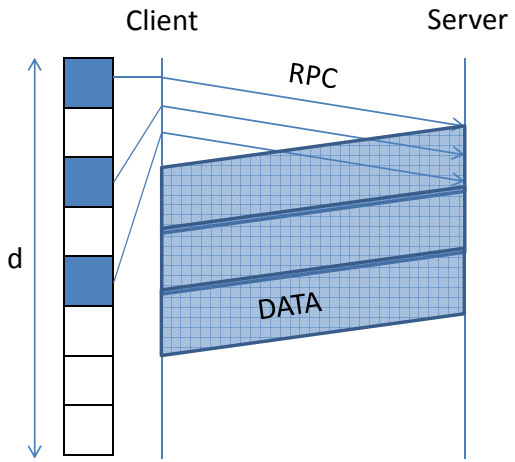

図 5 検出したパターンによる非同期転送
Fig. 5

このグラフから分かるように，いずれの回線遅延においても非同期型 RPC を用いたケースが高い性能を示していることが分かる．特に 25ms や 50ms の遅延においては，同期型 RPC を用いている場合には RPC バッファサイズを大きくし，シークされる領域も含めて転送しているため，シーク量に対する読み込み量の比率である 33%が性能の上限になっている．非同期型 RPC を用いることによりこの限界を超えて転送出来ていることが分かる．

5. まとめと今後の課題

本論文では，非同期型 RPC を活用し，シーケンシャルアクセスやストライドアクセスなどのパターンに対して，効果的に先読みを行う事による性能向上を，実際の性能評価を通じて示した．

一部の評価については，提案手法を完全に実装した上で行われたものでないため，アクセスパターンの検知などの要素については検証が不十分である．しかしながら，性能向上の可能性について一定の評価を示す事が出来た．

本論文の評価で用いたシステムは遠隔ファイルシステムとしての汎用性を備えているため，今後この点を活かして実アプリケーションの性能向上についても評価を行う予定である．

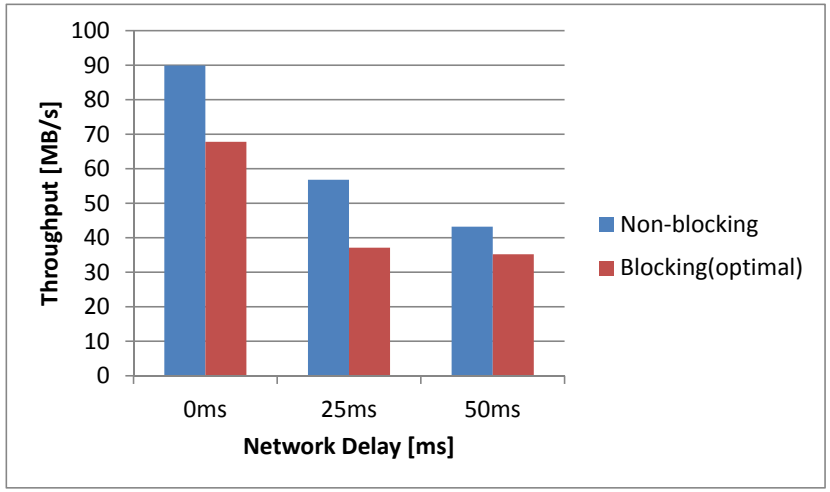

図 6 Non-blocking, Blocking RPC のストライドアクセスによる比較
Fig. 6

謝辞 本研究の一部は，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」および文科省次世代 IT 基盤構築のための研究開発「研究コミュニティ形成のための資源連携技術に関する研究」(データ共有技術に関する研究) 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) 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).
3) Callaghan, B., Pawlowski, B. and Staubach, P.: NFS Version 3 Protocol Specification, *RFC 1813* (1995).
4) Braam, P.J.: Lustre, *http://www.lustre.org/*.
5) Howard, J.H.: Scale and performance in a distributed file system, *ACM Trans. Computer Systems*, Vol.6, No.1, pp.51–81 (1988).
6) Satyanarayanan, M.: Coda: A Highly Available File System for a Distributed

Vol.2011-ARC-197 No.16
Vol.2011-HPC-132 No.16
2011/11/28

Workstation Environment, *IEEE Trans. Computers*, Vol. 39, No. 4, pp. 447–459 (1990).
7) Allcock, W.: GridFTP: Protocol Extensions to FTP for the Grid, *Global Grid Forum Draft* (2003).
8) : Fuse, *http://fuse.sourceforge.net/*.