

ロボットのクラウド化におけるディスク並列化

柳田将希 中野美由紀 菅谷みどり
芝浦工業大学

1. 研究の背景と目的

現在、人と自然なコミュニケーションを行うロボットが研究されており、こうしたロボットは会話で自然な時間で反応することが重要である。高度なロボットでは、高性能なマシンを用いることにより多くの処理を行なえるようにすることが一般的であり、アプリケーションにより画像解析や、声などの解析時にはハードディスクに保存してある過去のデータにアクセスすることが頻繁に行われる。こうした場合、ハードディスクが応答性能を低下させる要因となる。

ハードディスクの I/O に関する研究は過去に多くなされている。キャッシュによる応答性能向上が一般的であるが、必ずしも、キャッシュが十分でないケースや、キャッシュにヒットしないケースも十分ありうる[2]。予備実験を通じて、データ量が多いケースでは、キャッシュの制限による遅延の発生が顕著であったことから、キャッシュの制限があった場合でも、ハードディスクの並列化により速度向上が見込めるのかを明確にするものとした。本論文では検証を行った結果を報告する。

2. アプリケーションバッファの効果

2.1 実験内容

アプリケーション側で、効率的にメモリを利用することで性能改善が見込めるのかを明確にするために、アプリケーション側のバッファの調査をおこなった。実験環境を表1に示す。実験では `setvbuf()` を用いてバッファサイズを 4Kbyte から 1024Kbyte までを検証した。`setvbuf()` は入出力バッファを自動で決めたものではなく、任意のメモリ上に確保した領域を入出力バッファとして使用するようセットする関数である。この実験の時はキャッシュについては考慮していない。データサイズは 1024byte から 512Mbyte までのサイズとした。実験結果を図1、図2に示した。また、物理メモリ量の影響を調べるために、表2に示す 128GB の RAM を搭載したマシン構成により同様の実験を行った。

表1 実験環境

対象	詳細
CPU	Intel Core2Duo E4500
メモリ	DDR2-6400 1GBx1
ハードディスク	Seagate 7200.10 ST3250310AS 250GB 8Mbytecache
OS	Ubuntu14.4 64bit

表2 実験環境

対象	詳細
CPU	Intel XeonE5-2620 x2
メモリ	DDR3-10600 ECC Registered 8GBx16
ハードディスク	Seagate ST1000DM003

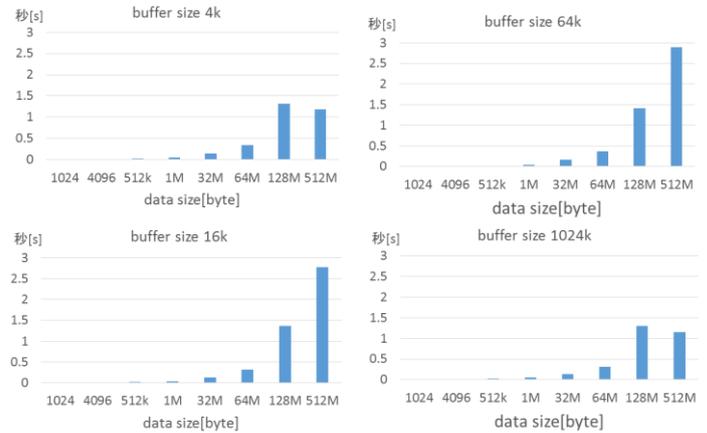


図1 バッファサイズ変更の影響

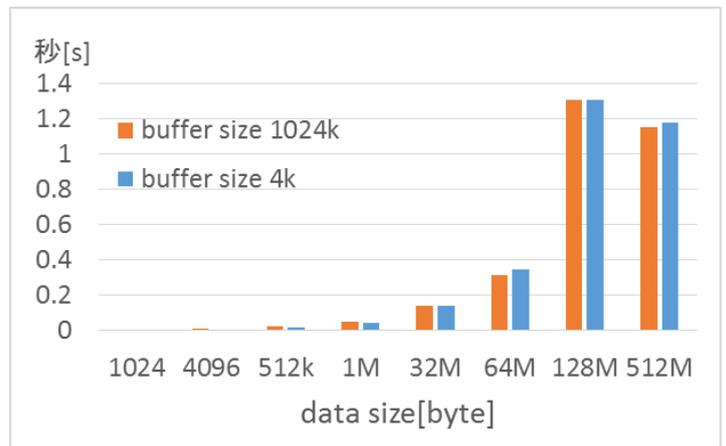


図2 バッファサイズ変更の影響

2.2 実験結果と考察

図1より、アプリケーション側のバッファサイズ変更の影響はほとんどなく、むしろ、性能には図2に示した全体のデータ量による影響があることが分かった。変化が現れたデータサイズは 128Mbyte であった。64Mbyte 以下ではそれぞれ誤差の範囲内に収まる読み込み時間が出た。

しかし、図1の左上や右下のようにデータサイズが 512Mbyte の時に逆に読み込み時間が下がってしまっている。表2の実験環境で同じ実験をしたところ buffer size 16Kbyte, 64Kbyte の場合でも buffer size 4Kbyte, 1024Kbyte でデータサイズ 512Mbyte で実験を行った時と同じ読み込み時間になった。このことから図2までの結果は実験に使用したマシンが 512Mbyte と小さい RAM しか持っていなかったことにより、キャッシュにヒットせず、ハードディスクからの読み込みが行われたことにより読み込みの時間が長くなったことが予測される。

3. ハードディスク並列化による性能向上

3. 1 実験内容

高速化にあたっては、OS 側のキャッシュの影響、また、ハードディスクの並列化の影響を比較するものとした。実験には予備実験で使用した表 2 のマシンを使用した。実験は二つの方法を比較するものとした。

実験 1：図 3 左のようにファイルを読み込むプロセスを 2 個生成しそれぞれ別のコアに割り当てハードディスク 1 台に対して同時にアクセスし、両方の読み込みが完了するまでの時間を測るものとした。

実験 2：図 3 右に表すとおり実験方法 1 と同じようにファイルを読み込むプロセスを 2 個生成し、それぞれ別のコアに割り当てる。今度はプロセスごとに別のハードディスクにアクセスするように設定する。

データはどちらの実験も 512Mbyte 分の文字列データである。fread() が一度に読み込むサイズも測定結果に影響するため、CentOS に標準で付属のディスクユーティリティのベンチマーク機能を用いて読み取り速度を計測した。

結果、160Mbyte/s 前後の読み取り速度があることがわかった。このことからこれより小さい数値の 128Mbyte を一度に読み込む数値として設定した。また、キャッシュにヒットした場合とどれだけの差が出るかも確認するためそれぞれの実験方法においてキャッシュあり、なしを測定した。キャッシュを無くす方法としてファイルを読み込む前にカーネルパラメーターを変更する sysctl コマンドを使って vm.drop_caches の値を 3 に書き換えることにより、ページキャッシュ、dentry、inode を解放している。dentry はキャッシュのファイル名、ディレクトリ構造、キャッシュ管理を行っている。inode はファイルサイズやモード情報などが格納されている。測定はそれぞれ 10 回行う。

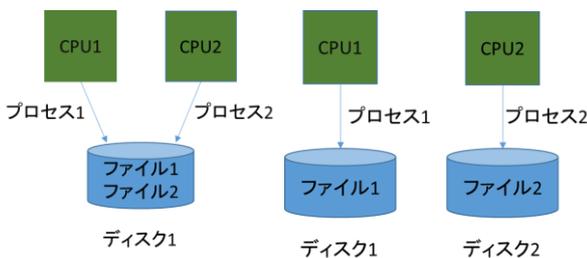


図3 並列アクセスの影響調査(実験1, 2)

3. 2 結果および考察

図 5 に本実験の測定結果の 10 回平均を示す。実験方法 1 では 512Mbyte x 2、計 1GB のデータを読み込むのに 5.93 秒かかった。対して、ハードディスク 2 台に並列で同時アクセスした実験方法 2 では 3.36 秒と大幅に時間を削減できている。キャッシュありの場合ではそれぞれで 2.49 秒、2.70 秒となった。

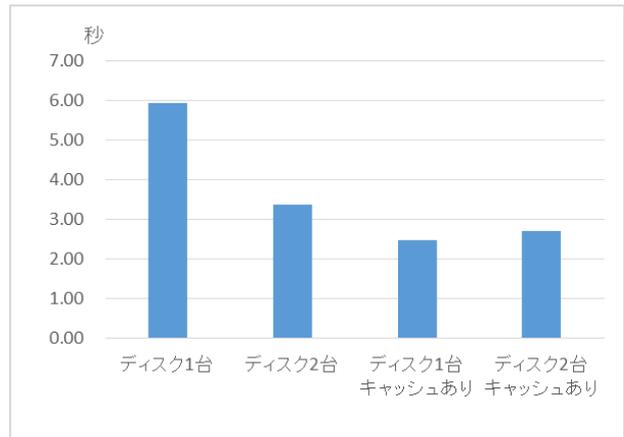


図5 実験1及び実験2の測定結果

結果としては、ハードディスク 2 台に対して並列に同時アクセスした場合にはハードディスク 1 台に対して同時アクセスした場合の約 57%の時間、約半分の時間で読み込み時間が済む。さらにキャッシュを用いた場合の時間に対して+約40%の時間であり、ハードディスク 1 台の場合の+約 130%から比べると大幅な改善となっている。ロボットのクラウド化においても大きなデータを扱うことを考えればこれだけの読み込み速度の差があるので応答性の向上が見込めるといえる。

4. まとめと今後の課題

本研究では、ハードディスクの性能改善に関する二つの調査結果を報告した。結果では、アプリケーションレベルでのバッファ操作は十分な成果が得られなかったが、ハードディスクの並列化による効果とキャッシュの効果の比較については、ある程度の成果が得られた。今後、必ずしもハードディスク容量に見合う物理メモリを準備することはできないことから、ハードディスクを増やすことで、安全性を確保しつつ速度を向上させる手法は必要であるといえる。ハードディスクのアクセス速度がメモリのアクセス速度を越えることはメモリ上でデータが操作されている以上ありえないことだが、今後ハードディスクの並列度を上げることにより読み込み速度の向上の限界点がどこかを探る必要がある。

また、読み込みにかかる時間がハードディスク 1 台に対して 2 台の場合でも完全に半分にならない。これらのオーバーヘッドについても調べる必要がある。

参考文献

- [1] 林宗佑, 楊世偉, 島田秀揮, 佐藤健哉: クラウドを利用した周辺環境情報に基づく高速自車位置推定手法の提案, マルチメディア, 分散, 協調とモバイル (DICOMO2013) シンポジウム
- [2] 高橋一志, 佐々木慎, 松宮遼, 大山恵弘: ページキャッシュ最適化におけるページのアプローチの検討, 情報処理学会研究報告. [システムソフトウェアとオペレーティング・システム]2013-OS-126(1), 1-9, 2013-07-24
- [3] 東京大学情報基盤センター黒田久泰 2006 「C 言語におけるファイル入出力の高速化」