平成20年度　　卒業研究論文

並列・投機書き込みを併用したファイルクラスタに関する検討

指導教員
松尾 啓志 教授
津邑 公暁 准教授

名古屋工業大学 情報工学科
平成17年度入学 17115054番

貴志 友哉

目 次

第1章

はじめに

近年、取り扱うデータ量の増加し続けている。このため、大容量で信頼性が高くスループットが高いストレージシステムが求められている。

従来のように１つのストレージ装置に書き込みしていては、そのストレージ単体以上のスループットを実現することはできない。そのため、複数のストレージクラスタを高速なネットワークで結合するネットワークファイルシステムや分散ファイルシステム実用化されている [1] [2]。

また、PC のプロセッサの性能向上と並列処理術の向上により複数の PC クラスタでストレージシステムを構成する分散ファイルシステムが実用化されている。これらは複数のストレージを組み合わせ、冗長性を持たせることで大容量で高い信頼性を得ている。

ファイルの容量が大きい場合にある１つのストレージ装置にのみ書き込むことは、そのストレージ装置に多大な負荷が掛かることになる。そのため、ある１つのストレージに負荷が偏る状況は好ましくないため、複数のストレージ装置に対して書き込みを行うほうが効率的である。

ストライピングのようにファイルを分割して複数のストレージ装置に書き込む方法として並列書き込み、投機書き込みがある。

本研究では、クライアント (システム利用者) に対して安定したスループットと高い可用性を提供するために並列書き込みと投機書き込みを併用するネットワークファイルシステムを提案する。また、提案手法はストレージクラスタへの通信において非同

期書き込みを行う．これにより通常の並列、投機書き込みを用いる手法よりもよりも高いスループットを得られることが期待される．

以降、第2章では関連研究を紹介する。第3章では本研究で用いるネットワークファイルシステムについて説明する。第4章では既存手法である並列書き込み、投機書き込み、非同期書き込みについて説明する。第5章で提案手法の説明を行う。第6章で実験を行い、最後に第7章で本研究のまとめを行う。

第2章

ネットワークファイルシステム

2.1　ネットワークファイルシステムの構成

本研究におけるネットワークファイルシステムについて述べる。ネットワークファイルシステムは図2.1に示すような、1台のメタサーバと複数台のI/Oサーバ、システム利用者であるクライアントから構成される。

各I/Oサーバとメタサーバは高速なネットワークで接続されているが、各I/Oサーバ同士は接続されていない。クライアントがシステムに対して出来ることとメタサーバとI/Oサーバの機能を以下に簡潔に示す。

クライアント

- メタサーバを介してシステムへ書き込みと読み込みを行うことができる
- メタサーバを通じてシステムが保持するファイルを知ることができる

メタサーバ

- ファイルのメタ情報を所持する
- クライアントとデータの送受信を行う
- I/Oサーバとデータの送受信を行う

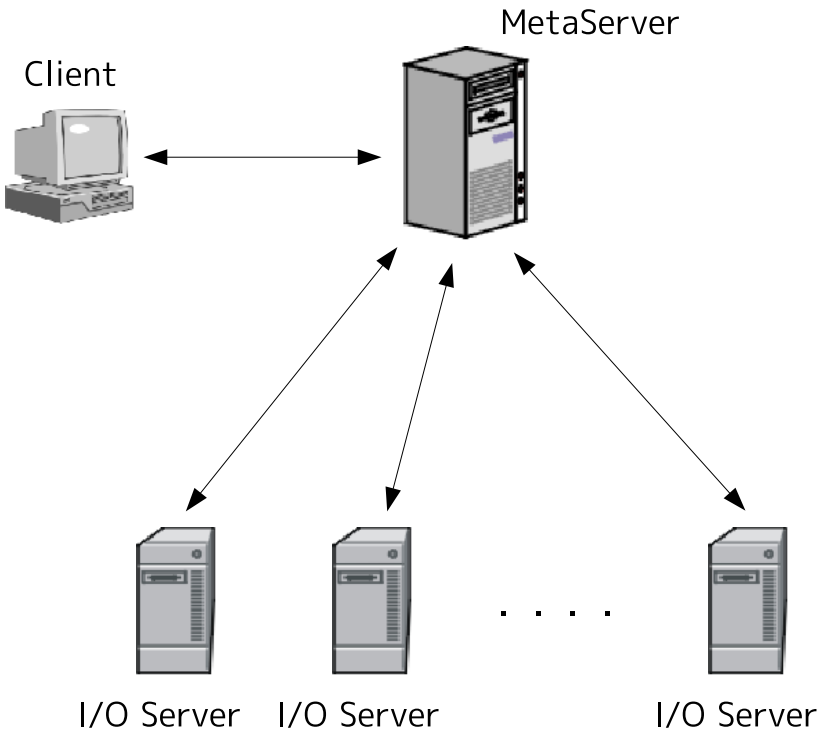

図 2.1: システムの構成

I/O サーバ

- メタサーバとデータの送受信を行う
- ストレージへ入出力を行う

各ファイルはブロックサイズ (一定の容量) で分割されて各 I/O サーバに保存されている。メタサーバはファイルのメタ情報を持つため、システムが持つファイルと各 I/O サーバが保持する分割ファイルを全て把握している。一方、クライアントはシステムが保持するファイルを知ることができるが、I/O サーバの位置が隠蔽されているため直接 I/O サーバへ分割ファイルの書き込み、読み込みはできない。このためクライアントはメタサーバを介してのみ I/O サーバへ書き込み、読み込みを行うことができる。

クライアントがファイルの書き込みを行う場合には、最初にメタサーバへファイルを送信する。メタサーバは受信したファイルを書き込む I/O サーバを決定し、決定し

たI/Oサーバへファイルを送信する。最後に、I/Oサーバは受けとったファイルをストレージへ書き込む。

ファイルの読み込みを行う場合には、最初にクライアントはメタサーバへ読み込むファイル名を通知する。メタサーバはファイル名から、ファイルのあるI/Oサーバへファイル名を通知する。I/Oサーバはストレージからファイルを読み出し、メタサーバへファイルを送信する。最後にメタサーバはI/Oサーバから受け取ったファイルをクライアントへ送信する。

第3章

ネットワークファイルシステムにおける従来の書き込み方式

本章では、ネットワークファイルシステムにおいて従来の書き込み手法である並列書き込み、投機書き込み、非同期書き込みについて説明する。

3.1 並列書き込み

並列書き込みでは、1つのファイルを複数のI/Oサーバへ分割して書き込みを行う方法である。最初にクライアントはメタサーバへファイルを送信する。メタサーバはファイルを$n(\geq 2)$分割して、分割したファイルを書き込むI/Oサーバを決定する。書き込むI/Oサーバが決定した分割ファイルをそのI/Oサーバへ転送する。

例として図3.1を用いて説明する。図3.1中の「File」が書き込み対象となるファイルである。また、「File」を一定の容量で分割したファイルを「A」,「B」,「C」としている。手順(1)として最初にクライアントはメタサーバへ「File」を送信する。手順(2)はメタサーバは「File」を書き込むI/OサーバをI/O Server1,I/O Server2,I/O Server3に決定する。3台のI/Oサーバへ書き込みを行うため、「File」を「A」,「B」,「C」に3分割する。手順(3)でメタサーバは「A」をI/O Server1,「B」をI/O Server2,「C」をI/O Server3へ送信し、I/Oサーバは受け取った分割ファイルをストレージへ書き込みを行う。それぞれI/O Server1,2,3は受け取った分割ファイルを書き込み完了後にメタサーバへ書き込み完了を通知する。メタサーバはI/O Serverから「A」,「B」,「C」

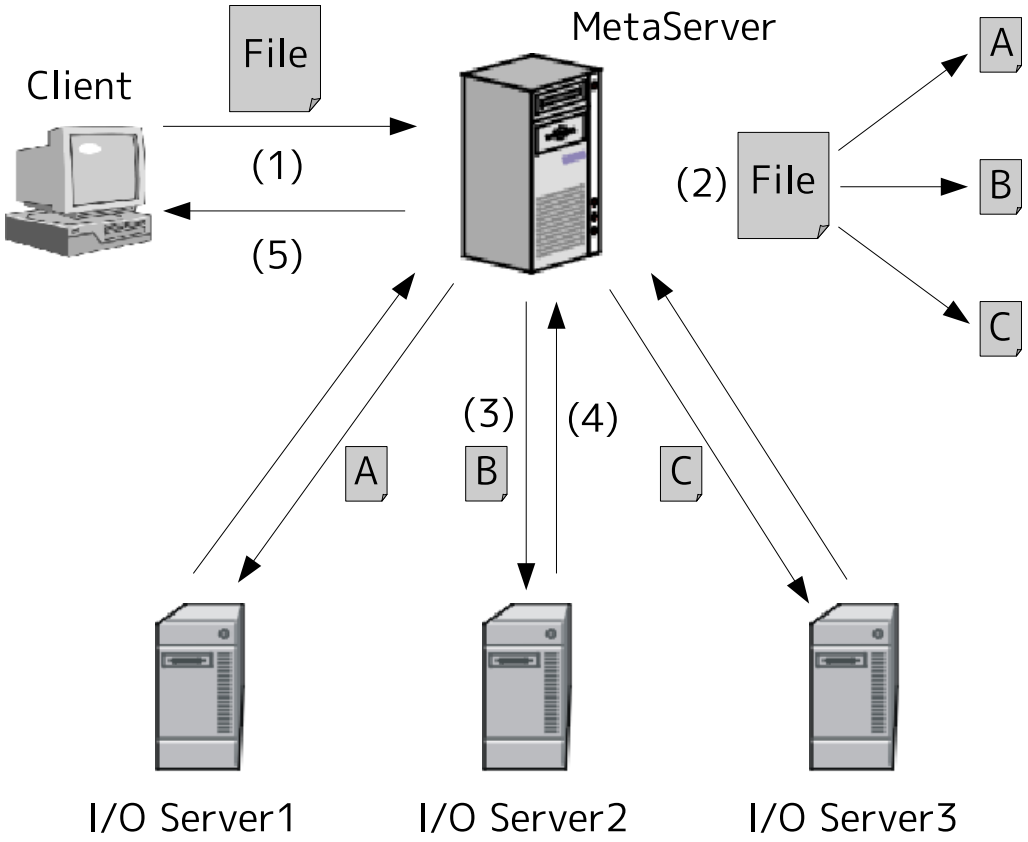

図 3.1: 並列書き込み

の分割ファイルの書き込み完了の通知を待つ。「A」,「B」,「C」の書き込み完了の通知を受け取り後、クライアントへ「File」の書き込みを完了を通知する。これにより「File」は 3 台の I/O サーバへ分割して保存されたことになる。

ファイルの読み込み時には、クライアントがメタサーバへ接続して読み込みたいファイル名を通知する。メタサーバは目的のファイルの分割ファイルのメタ情報から、分割ファイルを持つ I/O サーバを割り出す。分割ファイルを持つ I/O サーバから分割ファイルを受け取り、分割ファイルを統合する。最後に統合したファイルをクライアントへ転送する。

「File」を読み込む場合には、メタサーバが I/O Server1,2,3 より「A」,「B」,「C」を受け取る。次に「A」,「B」,「C」から「File」を復元してクライアントへ送信する。

並列書き込みは複数の I/O サーバへ書き込みを行うため、ある特定の I/O サーバへ

の書き込みが集中することを防ぐことができる。しかし、I/O サーバへの書き込みを行う時間は、そのI/O サーバの負荷により変動することになる。そのため、I/O サーバの中に極端に負荷が高いサーバがあれば、書き込みに時間が掛かり、スループットが低下することがある。つまり、並列書き込みにおけるスループットは一番高い負荷のI/O サーバへの書き込みに依存することになる。また、冗長性が無いため、あるI/O サーバが故障した場合には、ファイルへのアクセスができなくなる。さらに、I/O サーバが多いほど故障率が高い。

3.2 投機書き込み

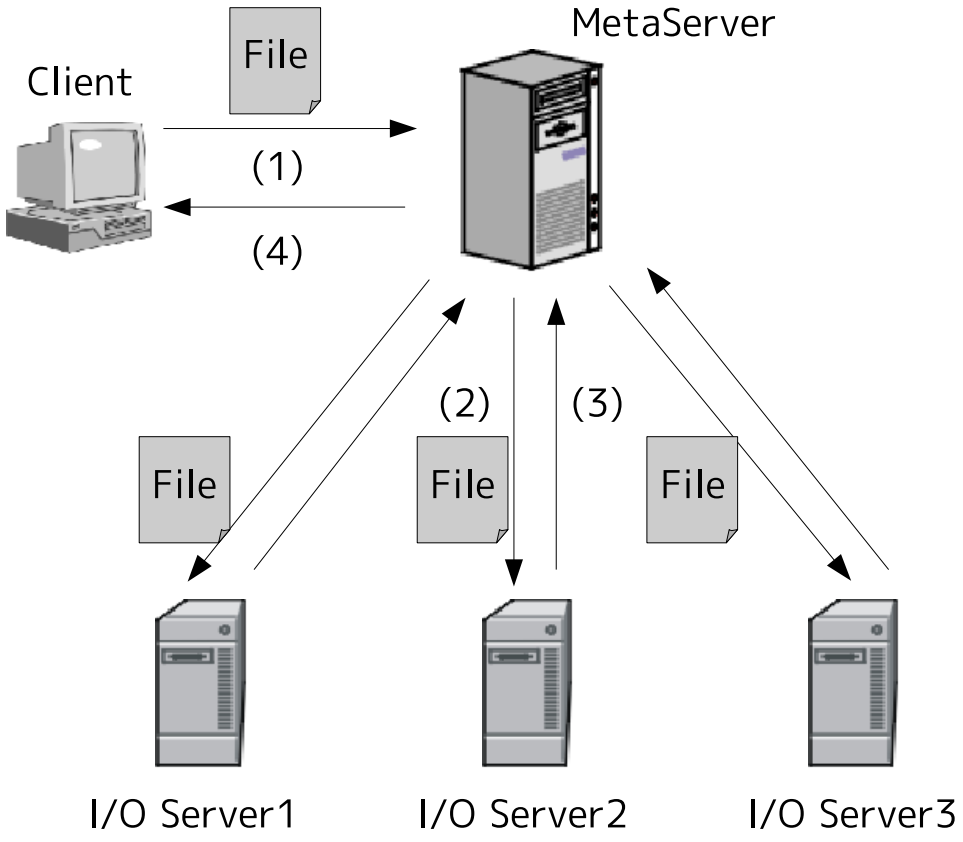

図 3.2: 投機書き込み

投機書き込みでは、ファイルを複数のI/O サーバに対して同時に書き込みを行う。

つまり、書き込み後には複数のI/Oサーバが同じファイルを所持することになる。書き込みの流れは、最初にクライアントはメタサーバへファイルを送信する。メタサーバは受け取ったファイルを書き込むI/Oサーバを$n(\geq 2)$台決定する。メタサーバは$n(\geq 2)$台のI/Oサーバへファイルを送信し、I/Oサーバは受け取ったファイルをストレージへ書き込みを行う。例として図3.2を用いて説明を行う。図3.1中の「File」が書き込み対象となるファイルである。手順(1)ではクライアントがメタサーバへ「File」を送信する。手順(2)はメタサーバが「File」をI/O Server1,2,3へ書き込むことを決定し、「File」をI/O Server1,2,3へ送信する。メタサーバはI/OServer1,2,3から「File」の書き込み完了の通知を待つ。手順(3)ではI/O Server1,2,3は「File」を受け取り、ストレージへ書き込みを行う。その後、メタサーバへ書き込み完了を通知する。最後に、手順(4)ではメタサーバがI/OServer1,2,3のいずれかから「File」の書き込み完了の通知を受け取り、クライアントへ書き込み終了の通知を出す。

複数のI/Oサーバに対して同時に書き込みを行うため、各I/Oサーバの負荷が低い状況でもスループットは書き込むI/Oサーバの台数に反比例して低くなる。あるI/Oサーバの負荷が極端に高い場合を考えると、そのI/Oサーバへの書き込みは時間が掛かる。しかし、他のI/Oサーバの負荷が軽いと、そのI/Oサーバへの書き込みが速く終わるためスループットの低下を防ぐことができる。つまり、投機書き込みにおけるスループットは一番負荷の軽いI/Oサーバへ依存することになる。また、同じファイルを複数のI/Oサーバが所持するため、１つのI/Oサーバが故障した場合でもそのファイルを読み込むことが可能であるので、冗長性が増し可用性が高い。その反面、ストレージの記憶容量が投機書き込みを行う台数×ファイルサイズとなるため効率が悪い。

3.3 非同期書き込み

非同期書き込みは、非同期通信によりファイルをI/Oサーバに転送し、書き込みを行う手法である。

非同期通信は送信者のデータ送信のタイミングと受信者のデータ受信のタイミングを合わせずに通信を行う通信方式である。同期通信ではデータ通信のリクエストを出

してからレスポンスが返るまで他の処理を行わないが、非同期通信ではレスポンスが返るまでの間に他の処理を行えるという利点がある。しかし、非同期通信を用いた場合には送信者は受信者が送信したデータを受け取っているかどうかを知ることができない。このため非同期書き込みには受信者が書き込みを完了したことを確認することが必要となる。

I/O サーバに書き込みを完了をしたことを確認するためにハッシュ関数を用いる方法がある。送信者は送信したデータをハッシュ関数によりハッシュ値 s を算出し、I/O サーバも受信したデータからハッシュ値 r を算出する。I/O サーバは送信者へハッシュ値 r を送信者へ送信し、送信者はハッシュ値 s,r を比較する。ハッシュ値が一致した場合に書き込みが完了したと判断する。

3.3.1 ハッシュ関数

ハッシュ関数とは与えられたデータから固定長の数値を生成する演算手法である。ハッシュ関数から得られた数値をハッシュ値と呼ぶ。ハッシュ関数の主な用途はデータの検証、改竄の検出、高速な検索である。データの検証のためには衝突困難性を満たすハッシュ関数を用いる必要がある。衝突困難性とは同じハッシュ値が容易に出力されないことである。

第4章

提案手法

既存の手法を用いた場合には、それぞれ利点と欠点があった。並列書き込みで書き込みを行う場合、欠点は冗長性が無いためストレージが故障した場合にファイルを読み込むことが出来ない。また、あるI/Oサーバの負荷が非常に高い場合には、そのI/Oサーバへの書き込みに時間が掛かるという欠点がある。投機書き込みを行う場合には、複数のI/Oサーバに同じファイルが保持される。あるI/Oサーバが故障しても他のI/Oサーバからファイルの読み込みができ、非常に高い可用性を持つが、複数へのI/Oサーバへ書き込みを行うため高速化は狙えない。以上のような利点と欠点を並列書き込みと投機書き込みは持っている。これらの欠点を補い合い、安定したスループットと高い可用性を提供するために並列書き込みと投機書き込みを併用した書き込みを提案する。１つのI/Oサーバに対して並列書き込みと投機書き込みを併用することは出来ない。そのため、併用するためにはI/Oサーバを図4.1のようにグループに分ける必要がある。グループに所属するI/Oサーバの台数は全て同じとする。各グループに対して投機書き込みを行い、グループ内の各I/Oサーバに対して並列書き込みを行うことで並列書き込みと投機書き込みを併用する。この手法により、並列書き込みと投機書き込みの欠点を補うことができる。

4.1　並列書き込み、投機書き込み、非同期書き込み

提案手法によるファイルの書き込みを説明する。この手法においてはI/Oサーバの各グループに対して投機書き込みを行い、グループ内のI/Oサーバへ並列書き込みを

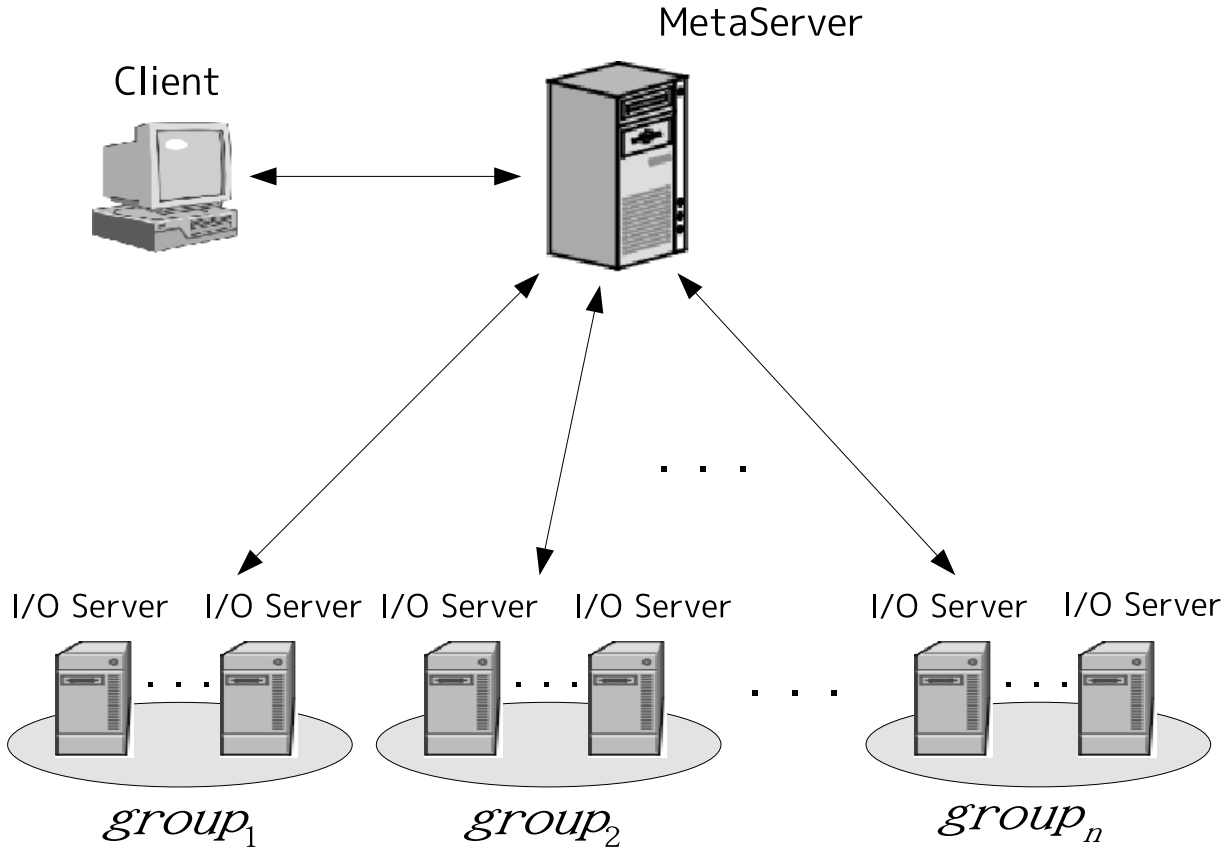

図 4.1: 提案手法の構成

行う場合と I/O サーバの各グループに対して並列書き込みを行い、グループ内の I/O サーバへ投機書き込みを行う場合の 2 通りがある。混乱を避けるために、前者を投機・並列書き込み、後者は並列・投機書き込みと定義をする。

図 4.2 の例を用いてファイルが投機・並列書き込みにより書き込まれる流れを示す。メタサーバはクライアントがシステムに対して書き込むファイル「A」を受け取る。次に、メタサーバは $group_i$ に所属する I/O サーバに対して並列書き込みを行う。

図 4.3 は分割ファイル「A-i」に注目して並列・投機書き込みによる書き込みの流れを示すものである。メタサーバはクライアントよりファイル「A」を受け取り、「A」をブロックサイズで分割する。分割ファイル「A-i」を $group_i$ へ投機書き込みを行うことを決定する。次に、メタサーバは $group_i$ に所属する全ての I/O サーバへ「A-i」を送

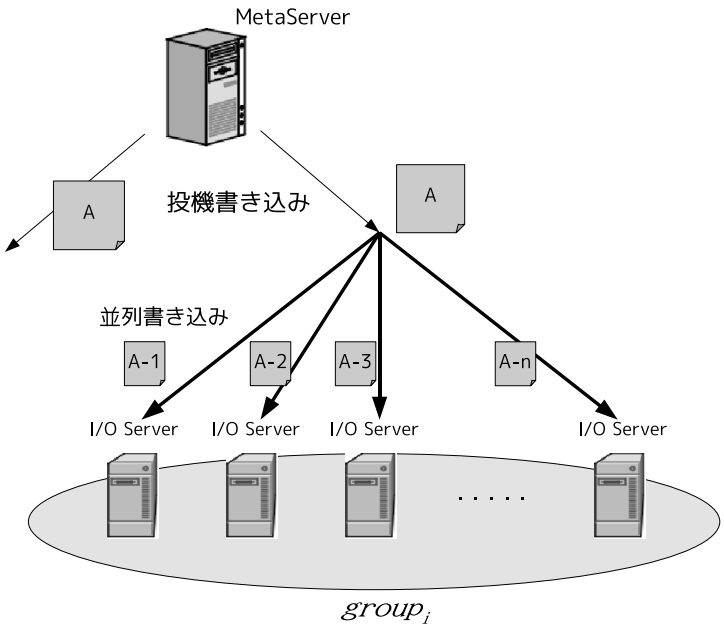

図 4.2: 投機・並列書き込み

信する。$group_i$ の I/O サーバはそれぞれのストレージへ「A-i」を書き込み、「A-i」の書き込みが完了したことをメタサーバへ通知する。メタサーバは $group_i$ に所属するいずれかの I/O サーバから「A-i」の書き込み完了の通知を受けた時点で「A-i」の書き込みが完了したと判断する。これを全ての分割ファイルに対して行う。

投機・並列書き込み、並列・投機書き込みは伴に、高い可用性と高いスループットを提供することができる。また、投機・並列書き込み、並列・投機書き込みに非同期書き込みを用いることでより高いスループットを得ることが期待できる。非同期書き込みを用いる場合にはメタサーバは I/O サーバへの書き込みを確認するためにハッシュ関数を用いる。そのため、非同期書き込みを用いる場合の書き込み完了を確認するまでの時間はハッシュ関数によるオーバヘッドにより、同期書き込みによる書き込み完了時間よりも時間が掛かる。

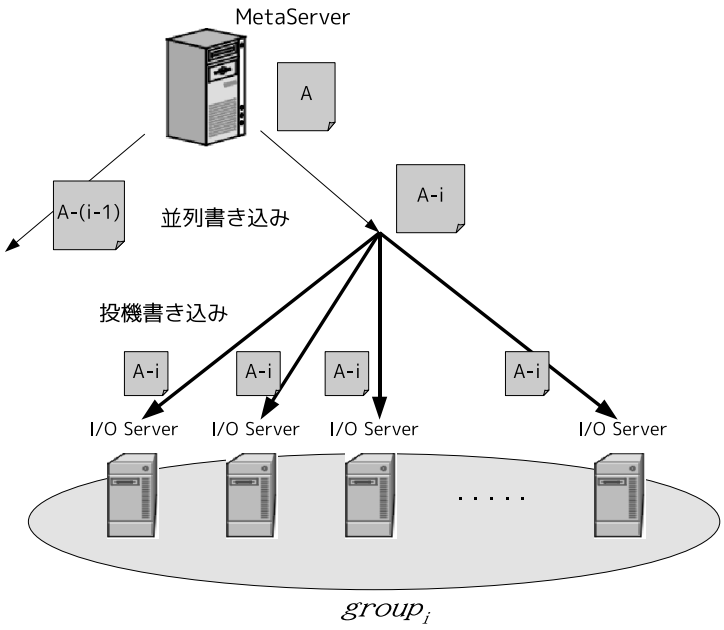

図 4.3: 並列・投機書き込み

4.2 I/O サーバのグループ化

I/O サーバのグループ化は I/O サーバの数により様々な組合せがある。例えば I/O サーバが 6 台ある場合には、I/O サーバ 3 台で 1 グループを形成し 2 グループが出来る組合せと I/O サーバ 2 台で 1 グループを形成し 3 グループを形成する組合せの 2 通りがある。投機・並列書き込みを用いて書き込みを行い、I/O サーバが 6 台ありグループが 2 つある状況で 2 つの I/O サーバに重い負荷が掛かっているとする。負荷のある I/O サーバ 2 台が同じグループに所属している場合には、もう 1 つの負荷のある I/O サーバを持たないグループの書き込みは速く終わる。しかし、負荷のある I/O サーバが 1 グループに 1 台所属している状態では、どちらの書き込み時間も遅くなってしまうような状況が起きる。このように、大きな負荷がある I/O サーバがどのグループに所属するかでスループットが大きく変化するような状況も起こりうる。

第5章

評価

第3章で述べたネットワークファイルシステムを用いてI/Oサーバの負荷を変動させ、並列書き込みと投機書き込みのスループットの関係について評価した。

5.1 実験環境

実験は全て実機で行った。クライアント、メタサーバ1台、I/Oサーバ5台の計算機環境を表5.1に示す。

クライアントが1,5,10,15(Mbyte)の容量のファイルをネットワークファイルシステムへ書き込みを行い、メタサーバからファイルの書き込み完了の通知を受け取るまでの時間を計測した。また、ブロックサイズは1Kbyteとした。今回の実験では3台のI/Oサーバの端末上で100Mbyteのデータをファイルに書き込むプログラムを動作させ続けて負荷を与えた。

	クライアント,I/Oサーバ	メタサーバ
CPU	Pentium 4 CPU 3.00GHz	AMD Phenom(tm) 9600 Quad-Core 1.15GHz
メモリ (byte)	1G	8G
回線速度 (bps)	100M	100M

表 5.1: 計算機環境

5.2 結果と考察

実験により得られたデータを図 5.1 へ示す。まず、負荷がない場合の並列書き込みと投機書き込みのデータを見ると常に投機書き込みのスループットが低いことがわかる。これは 5 台の I/O サーバへ書き込みを行っているためにスループットが並列書き込みよりも低い。

負荷をかけた場合と負荷がない場合の書き込み時間に着目すると、並列書き込みでは負荷をかけた場合、負荷がない場合よりも平均 613ms 程書き込み時間が増えていた。投機書き込みでは、負荷がある場合もない場合も書き込み時間にほとんど変化が無かった。並列書き込みは、負荷をかけた I/O サーバへの書き込みに時間がかかりクライアントへ書き込み完了を通知するのに時間がかかった。投機書き込みについては、負荷のかかっていない I/O サーバへの書き込みが負荷のかかった I/O サーバへの書き込みより速く終わったため、どの I/O サーバにも負荷がない場合と書き込み時間が変わらない。

負荷がある場合の 1Mbyte のファイルの書き込み時間に着目すると、負荷がない場合には見られなかった投機書き込みよりも並列書き込みに時間がかかった。このことから、I/O サーバの負荷の高さにより投機書き込みより並列書き込みのほうがスループットが悪いことがあるということが評価された。

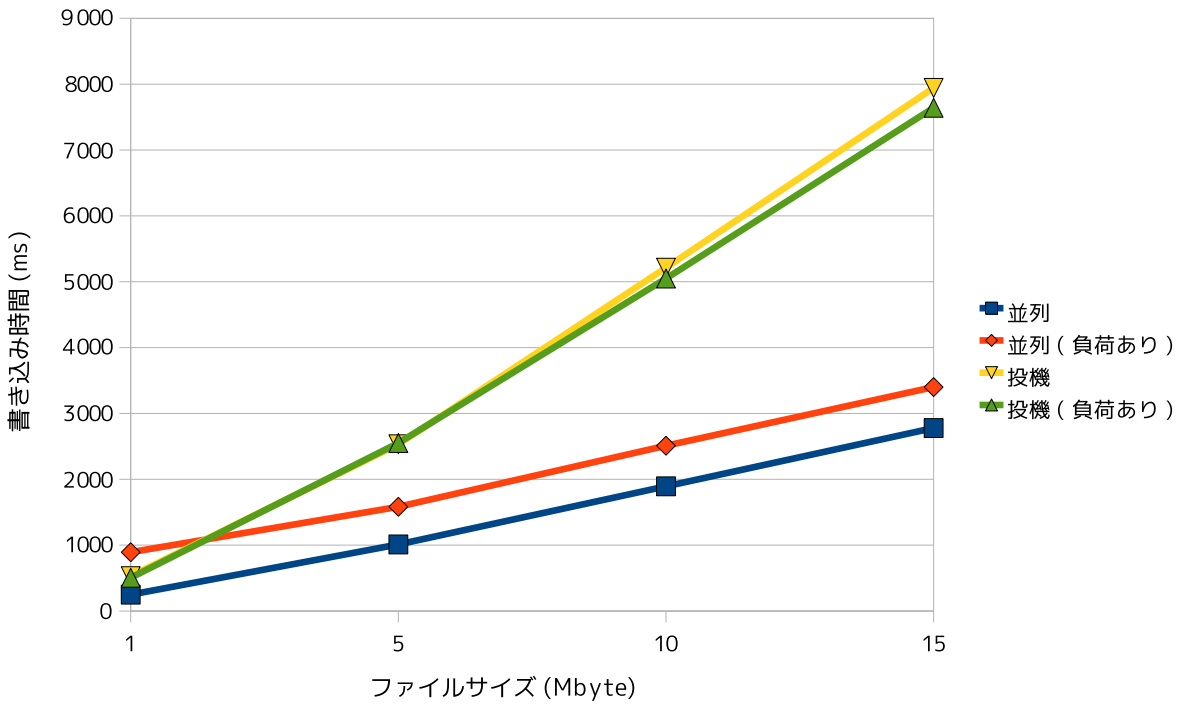
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
書き込み時間 (ms)
1
5
10
15
ファイルサイズ (Mbyte)
並列
並列 (負荷あり)
投機
投機 (負荷あり)

図 5.1: 各書き込み時間

第6章

まとめ

本研究では、クライアントに対して安定したスループットと高い可用性を提供するために並列書き込みと投機書き込みを併用したネットワークファイルシステムを提案した。また提案手法ではスループットの向上のためにストレージクラスタへの通信に非同期書き込みを導入した。投機書き込みと並列書き込みのスループットがI/Oサーバの負荷の状況においてどのように変動するかについて、提案手法の試作システムを実装し、実験により評価を行った。

I/Oサーバのグループ化、負荷変動への対応、投機書き込みの効率化のためのメタサーバ・I/Oサーバ間のブロードキャストの導入は今後の課題である。

参考文献

[1] S. Ghemawat, H. Gobioff, and S.-T. Leung “ The google file system, ”
ACM SOSP, October 2003.

[2] Isilon IQ Clustered Storage, http://www.isilon.com/.

[3] D.Patterson, G.Gibson, and R.Katz,“ A Case for Redundant Array of Inexpen-
siveDisks(RAID), ”Proc.of ACM SIGMOD’88,pp.109-116,June 1988.

謝辞

本研究のために多大な御尽力を頂き、日頃から熱心な御指導を賜った名古屋工業大学の松尾啓志教授に深く感謝致します。

また、本研究の際に多くの助言、協力をして頂いた齋藤彰一准教授、津邑公暁准教授、松井俊浩助教及び松尾・津邑研究室ならびに齋藤研究室の皆様に深く感謝致します。