プライベートクラウド環境における分散ファイルシステムの試作
―メタデータ管理のクライアントへの委譲―

2009SE016 坂 亮平　2009SE106 加藤 駿一
指導教員:宮澤 元

1　背景

近年，クラウドコンピューティングが注目を集めている．クラウドコンピューティングとはインターネットなどのコンピュータネットワーク経由でサービスを提供するコンピュータの利用形態である．クラウドコンピューティングはパブリッククラウドとプライベートクラウドに大別される．パブリッククラウドでは, 複数のサーバがインターネット経由で接続されている一般ユーザに対してサービスを提供するのに対し，プライベートクラウドでは，組織内のシステムを利用して企業等の特定の組織内のみでサービスの提供を行う．

プライベートクラウドには以下の特徴がある．

- リソースに制限がある．
  組織内の環境やコストの制限から, 利用できるリソースに制限ができる．
- クライアントに構造がある．
  パブリッククラウドと違い部門ごとのグループとしてクライアントをとらえることができる．
  グループごとに必要なサービスやランダムアクセスのパターンが異なると考えられる．
- セキュリティ
  利用者が組織内に限られるので，重要な情報を組織内のみにとどめ，セキュリティポリシの実現が期待できる．また, クライアントもサーバと同様にシステムの管理範囲に置かれる．

我々はプライベートクラウドの特徴を踏まえ，プライベートクラウドにおいて効率的に動作する分散ファイルシステムを開発している．本分散ファイルシステムではサーバ機能の一部をクライアントに委譲することによりサーバの負荷を低減し，リソースの有効利用を図る．

本稿では本分散ファイルシステムのメタデータサーバについて述べる．本メタデータサーバはクライアントへメタデータ管理の一部を委譲することでメタデータ管理の負荷をサーバホストからクライアントへ分散することができる．実装したメタデータサーバを用いて実験を行い，スレーブサーバの有効性を確認する．

2　システムの全体像

本節では, 本分散ファイルシステムの全体像を示す.

2.1　分散ファイルシステム

本分散ファイルシステムはクライアント側システム，ストレージサーバ，メタデータサーバから成る．図 1 にファイルシステムへのアクセスの全体像を示す．システムコールを受け取ったクライアント側システムがメタデータサーバに PATH 名を問い合わせ，対象のファイルについての inode 番号などのメタデータをメタデータサーバから得る．クライアント側システムはこの情報をもとにストレージサーバに接続し，対応するファイルデータを操作する．

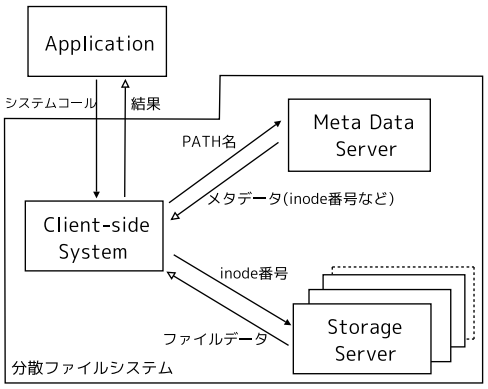

図 1　分散ファイルシステムの全体像

3　メタデータサーバの概要

本節では本メタデータサーバの設計と動作について述べる．

3.1　メタデータサーバの機能

本メタデータサーバの機能はメタデータ管理とそのクライアントへの委譲，ならびにクライアントのファイルキャッシュの一貫性保持である．

3.1.1　メタデータ管理

メタデータサーバはファイルシステム全体の名前空間を管理すると共にファイルごとのメタデータを管理する．クライアント側システムからの open や close などの要求に対応してメタデータ操作を行う．名前空間はディレクトリ階層で管理される．ファイルシステム内でデータを追加，削除，移動するときはこの名前空間を操作し，ファイルシステムに反映させる．

メタデータとしてはファイルの inode 番号，PATH 名，サイズ，最終更新日時などが記録される．ファイルの inode 番号はメタデータサーバが割り当て，ストレージサーバはこの inode 番号を用いて保存ストレージの決定などのファイルデータの管理する．

3.1.2　メタデータ管理の委譲

メタデータ管理の一部をクライアントホストに委譲することでメタデータサーバの負荷を低減させることができる．メタデータ管理の委譲を行うためにクライアントホストで稼働するスレーブサーバを使用する．これに対して, ファイルシステム全体のメタデータの管理をするサーバをマスターサーバと呼ぶ.

3.1.3 クライアント側のキャッシュ管理

クライアントホストはファイルデータをストレージサーバから読み込むときにそのファイルをキャッシュする場合がある．メタデータサーバはクライアントホストが持つキャッシュの正当性を保証するとともにキャッシュの一貫性保持を行う．

3.2 メタデータサーバの構成

メタデータサーバはマスターサーバとスレーブサーバで構成される．マスターサーバは一つのクラスタにつき一つのホストで稼働する．スレーブサーバはクライアントホストで稼働し，指定された特定のディレクトリのメタデータを管理する．

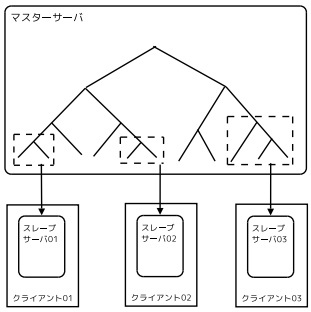

図 2 メタデータ管理のクライアントホストへの委譲

図 2 はメタデータ管理のクライアントへの委譲を図に示したものである．マスターサーバはファイルシステム全体のディレクトリツリーのメタデータを管理する．各クライアントホストで稼働するスレーブサーバには，一部の範囲のディレクトリツリーの管理が委譲される．

3.2.1 マスターサーバとスレーブサーバ

マスターサーバはファイルシステム全体のメタデータの管理をすると共に，スレーブサーバの管理を行う．マスターサーバは一つのホストで稼働するので，処理が集中する．スレーブサーバには, マスターサーバにかかる負荷を低減させる目的がある．例えば, あるユーザの特定のディレクトリへのアクセスが他のユーザに比べて多い場合，このディレクトリのメタデータの管理をそのユーザのホストで行えるようにスレーブサーバにメタデータ管理を委譲することで，そのユーザはそのディレクトリへのアクセスを効率的に行うことが可能となる．これによりマスターサーバにかかる負荷をクライアントに分散することが期待できる．

図 3 はスレーブサーバによりメタデータ管理をクライアントホストへ委譲した時のクライアントの通信を図にしたものである．Client A はスレーブサーバを持っているクライアントホストである．

(1) はクライアント側システムとマスターサーバとの基本的な通信である．

(2) はクライアント側システムとスレーブサーバの通信である．クライアント側システムは自身と同じホスト上のスレーブサーバと接続することでマスターサーバと接続することなくメタデータの操作が可能となる．

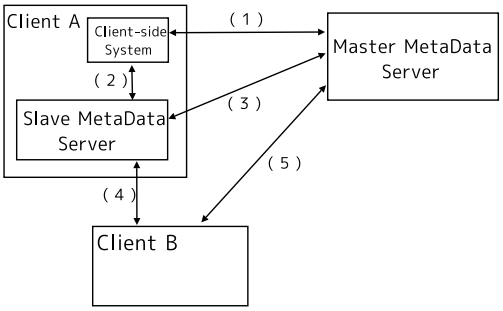

図 3 マスターサーバとスレーブサーバの通信

(3) はマスターサーバとスレーブサーバの通信であり，スレーブサーバへメタデータのコピーを行う時やスレーブサーバのメタデータに変更が起きたときにデータの同期をとる時にこの通信が用いられる．

(4) は自分以外のクライアント側システムからのスレーブサーバへのアクセスである．

(5) は (1) と同様にクライアント側システムからマスターサーバへのアクセスだが，このアクセス時に指定されたファイルのメタデータ管理がスレーブサーバに委譲されていれば，(4) のようにアクセスをリダイレクトさせその処理をスレーブサーバに行わせる．

3.3 メタデータサーバの動作

本メタデータサーバの動作について述べる．

3.3.1 スレーブサーバの起動

スレーブサーバは現状では，クライアントホストに委譲されるディレクトリを手動で入力し起動する．スレーブサーバは起動する時にマスターサーバから委譲されるディレクトリの PATH 名とともにそのディレクトリのメタデータを受け取る．スレーブサーバは受け取ったメタデータを保存し起動完了となり，クライアント側システムからのアクセスの受付を開始する．

3.3.2 クライアント側システムからのアクセス

マスターサーバは全てのクライアント側システムからのアクセスを受け付けるが，指定された PATH 名によりその処理をマスターサーバで行うかスレーブサーバで行うか決める．スレーブサーバに委譲しているディレクトリへアクセスが来た場合は，その処理をスレーブサーバにリダイレクトする．

クライアント側システムは基本的にはマスターサーバへのアクセスを初めに行う．この時にマスターサーバは指定された PATH 名をスレーブリストと参照し，マスターサーバとスレーブサーバのどちらが処理を行うか判断する．スレーブリストとは，現在稼働しているスレーブサーバのホスト名と委譲されているディレクトリが記録されたリストである．クライアント側システムから要求された PATH 名がどのスレーブサーバにも委譲されていなければ，マスターサーバが処理する．スレーブサーバに委譲された範囲であればマスターサーバはクライアント側システムにスレーブサーバのホスト名を返す．

要求を送ったクライアント側システムがマスターサーバからスレーブサーバのホスト名を受け取った場合，そのクライアント側システムはスレーブホストリストにホスト名とディレクトリ名を記録する．スレーブホストリストはそのクライアント側システムがアクセスをしたことがあるスレーブサーバのホスト名のリストとなる．

クライアント側システムは次回からメタデータサーバにアクセスするときにまずスレーブホストリストを参照し，PATH 名が一致すれば直接スレーブサーバへアクセスする．図 4 にクライアント側システムからのメタデータサーバへのアクセスの分岐を示す．

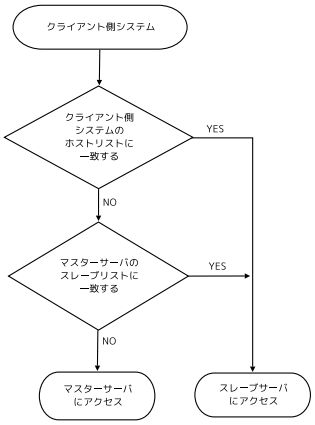

図 4 メタデータサーバへのアクセス

3.3.3 メタデータの同期

スレーブサーバとの通信時やクライアントに障害が起きたときにデータの損失を防ぐため，マスターサーバとスレーブサーバのメタデータに一貫性を持たせなければならない．そこで，スレーブサーバに委譲された範囲のメタデータに更新が起きた場合に同期処理を行う．同期処理は以下のように実行される．まずスレーブサーバはクライアント側システムからの要求を受ける．要求を受けたスレーブサーバはその要求の実行結果を返す．クライアント側システムからの要求の処理を行った時にメタデータに変更があればマスターサーバにアクセスし，変更したメタデータの内容をマスターサーバへ送ることでメタデータの同期を行う．

3.3.4 スレーブサーバの停止

スレーブサーバが停止する時は，クライアントホストがダウンするかオフラインになる場合である．スレーブサーバが停止した場合すぐにマスターサーバへの通知は行われない．停止したスレーブサーバに委譲していたディレクトリへのアクセスが次回，他のクライアント側システムにより行われたとき，そのアクセスは失敗となる．このとき，このクライアント側システムは自身のスレーブホストリストの該当エントリを無効化してから，マスターサーバへのアクセスを行う．このとき，クライアント側システムはスレーブサーバが無効であったというフラグを付けてアクセスを行う．マスターサーバは停止したスレーブサーバに無効マークを付ける．無効マークを付けられたスレーブサーバへのリダイレクトは次回以降から行われなくなる．図 5 はスレーブサーバがオフラインの場合の通信の流れを図に表したものである．オフラインのスレーブサーバへの通信が初めて行われたときにこの図のようになり，次回からはこのスレーブサーバへのアクセスは行われなくなる，

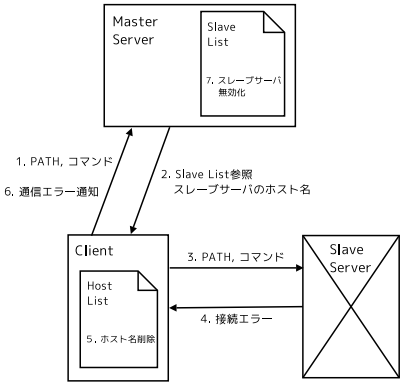

図 5 スレーブサーバがオフライン時の通信

3.3.5 スレーブサーバの再開

スレーブサーバが停止した状態から再び起動を始めるとき，それまで持っていたメタデータを全て破棄する．破棄が完了したら，最新のメタデータをマスターサーバから受け取る．これはスレーブサーバが古いメタデータを使うことを防ぐためである．スレーブサーバが再開するときに，マスターサーバはスレーブリストからそのスレーブサーバに付けられた無効マークを外す．無効マークを外されたスレーブサーバはクライアント側システムからのアクセスの受付が可能な状態になり再開が完了する．

3.4 キャッシュの一貫性保持

ファイルのメタデータには，そのファイルのキャッシュを持っているクライアントホストのリストが含まれる．メタデータサーバは，ファイルに更新が起きた場合，更新されたファイルの更新前のキャッシュを持っている全てのクライアントホストに対してキャッシュの削除命令を送信する．これは AFS[1] の callback と同様の処理である．

クライアントホストがオフライン状態であり，削除命令を受け取ることができなかった場合，そのクライアントホストが再度オンラインになり，次回キャッシュを読み込む時，そのキャッシュが最新であるかメタデータサーバに問い合わせを行う．この方法を用いることによりクライアント側システムは古いキャッシュを利用することはなくなる．

4 実験

スレーブサーバを用いることにより，マスターサーバのみの時と平均応答時間，マスターサーバへの負荷がどう変化するかを調べるために実験を行う．またアクセスするメタデータの状態によりどう変化するかを明らかにする．クライアント側システムが open から close までの処理を 100 回繰り返しメタデータアクセスを行い，その

平均応答時間を計測する．またそのときのマスターサーバの最大 CPU 利用率を計測する．

メタデータの状態とは，(1) スレーブサーバを用いるか，(2) スレーブサーバはクライアントと同ホスト (自スレーブ) か他ホスト (他スレーブ) か，(3) 有効なキャッシュを持っているか，(4)close 時にメタデータの更新を行うか，これらの組合せによるものとする．

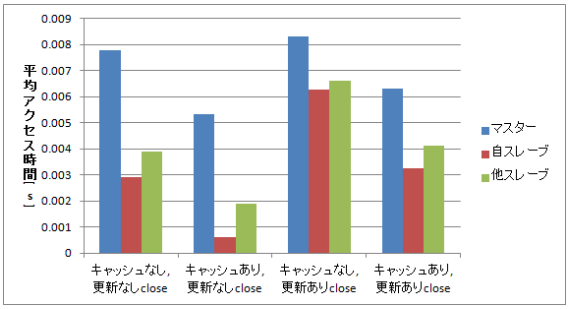

図 6　スレーブサーバの有無による平均応答時間

図 6 は平均応答時間をグラフにしたものである．マスターサーバのみと通信を行う場合は，有効なキャッシュを持っていれば応答時間は短くなり，close 時に更新があれば応答時間が長くなった．スレーブサーバを用いる場合は，同ホストのスレーブサーバにアクセスする方が他ホストのスレーブサーバへアクセスするよりも応答時間が短くなる，マスターサーバのみのアクセスよりも，スレーブサーバを用いたアクセスの方がアクセス時間が短くなっていることがわかる．

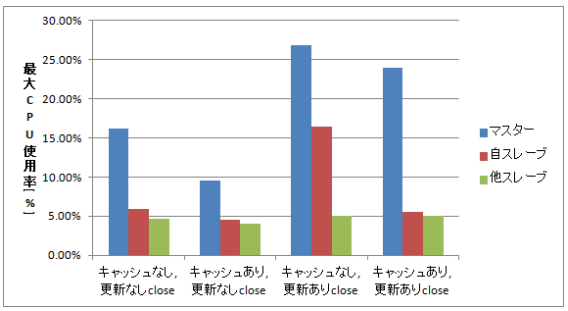

図 7　スレーブサーバの有無による CPU 負荷

図 7 はマスターサーバの最大 CPU 使用率をグラフにしたものである．実験の結果からスレーブサーバを用いれば，いずれの場合でもマスターサーバの負荷を軽減できることがわかる．

5　考察

メタデータに変更がない場合は，スレーブサーバは同期処理を行わない．よってスレーブサーバとマスターサーバで処理内容にほとんど差はないと考えられる．この場合は同ホストのスレーブサーバへのアクセスが一番早くなると考えられ，実験結果からも確認できる．他スレーブサーバへアクセスする場合は，マスターサーバのみの時とほぼ同等であると考えられるが，実験結果からはいずれの場合でもスレーブサーバを用いた方が応答時間が短くなった．この原因として考えられるのは，マスターサーバのみへのアクセスではメタデータがスレーブサーバに委譲されていないか確認する処理があることである．メタデータに変更がある場合は同期処理が増えるため，スレーブサーバを用いると応答時間が長くなると考えられる．しかし実験結果からはいずれの場合もスレーブサーバを用いた方が処理が早くなった．原因として考えられるのは更新なしの時と同じようにスレーブサーバへの委譲の確認処理が考えられる．

6　関連研究

Ceph[2] は数テラから数ペタバイトのデータを想定して作られた大規模分散ファイルシステムである．ストレージクラスタ，メタデータサーバ，クラスターモニタによって構成される．メタデータサーバは数十から数百台に分散される．メタデータサーバは一つのサーバに処理が集中するのを避けるために複数に分散している．複数のメタデータサーバがファイルシステムの名前空間を部分的に管理する．各サーバがどの部分を管理するかは，作業負荷に適応して均等になるように動的に決められる．我々のシステムではクライアントの使用状況に対応しながらメタデータ管理の負荷をクライアントに分散させることを目的としている．

7　まとめ

本研究ではメタデータ管理をクライアントへ委譲させるメタデータサーバの開発を行った．マスターサーバとスレーブサーバを用いた実験の結果から，スレーブサーバを用いることでマスターサーバの負荷をクライアントに分散することができた．

今後の課題には，本メタデータサーバの実用性を高めることがあげられる．そのために，クライアント側システムからのアクセスパターンを解析し，適応しながらスレーブサーバに管理を委譲させることが必要である．マスターサーバが自動でスレーブサーバの起動を行うことで本システムを効率的に動作させることができると考えられる．

参考文献

[1] John H. Howard, Michael L. Kazar, Sherri G. Menees, David A. Nichols, M. Satyanarayanan, Robert N. Sidebotham, and Michael J. West, “Scale and Performance in a Distributed File System, ” in *ACM Transactions on Computer Systems (ACMTOCS)*, pp. 51-81, Volume 6, Number 1, February 1988.

[2] Sage A. Weil, Scott A. Brandt, Ethan L. Miller, and Darrell D. E. Long, “Ceph: A Scalable, High-Performance Distributed File System,” in *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2006)*, pp. 307-320, 2006.