広域分散環境における分散ファイルシステム Hadoop の遠隔データアクセス制御手法の提案と評価

理学専攻 情報科学コース　百瀬 明日香

1　はじめに

情報爆発の現代におけるデータ処理の負荷やストレージコストの問題に対して，汎用機器を用いて高度な集約処理を行う分散ファイルシステムに注目が集まっている．本研究では広域分散環境において，近接/遠隔アクセスネットワークを併用することで大規模なデータ損失にも対応できる信頼性の高い分散ファイルシステムの運用に着目した．こうしたシステムの運用では，遠隔地へのアクセスによるネットワーク遅延がデータ処理効率に影響を及ぼすことが分かっている [1]．そこで本研究では遠隔ノードへのデータ配置制御手法を提案および実装し，遠隔データアクセス制御を用いた分散ファイルシステムの性能向上手法についての検討を行った．分散ファイルシステムの実装としてはオープンソースソフトウェア Hadoop Distributed File System（以下 HDFS）[2] を使用した．

2　実験環境

クラスタ自動構築・管理ツール Rocks[3] を用いて構築したローカルクラスタ上に Hadoop-0.20.2 をインストールし，Hadoop クラスタを構築した．マシンスペックは表 1 に示す通りである．また分散されるファイルの最小単位であるブロックサイズは 2.0MB とし，測定のためのベンチマークには，Hadoop に付属の TestDFSIO プログラムと Map/Reduce 処理をメインに行うベンチマークである自作の転置インデックスプログラムを使用した．

転置インデックスプログラムでは URL とテキスト 50,000 行からなる入力データを用意し，Map 処理として文書を N-gram モデルを用いて切り出し，URL をキー，含まれるテキストをバリューとして抽出する．中間処理で（キー，バリュー）データをバリューでソートし，Reduce 処理で重複したキーの削除を行う．なお，N-gram により切り出す文字列の長さは n=5 の場合を採用した．

3　高遅延環境における性能測定

3.1　実験概要

人工遅延装置 dummynet を使用して高遅延接続を含む模擬的な広域分散環境を構築し，Namenode1 台と Datanode3 台のうち Datanode1 台が遠方に存在すると仮定した環境での測定を行う（図 1）．ここで Namenode とはファイルのメタデータやクラスタ内のノード管理を行うノード，Datanode は実際にデータが格納されるノードである．分散されるファイルのレプリカ数は 2 とした．

表 1: マシンスペック

	OS	CPU	Main Memory
Master node	Linux 2.6.9-55.0.2. Elsmp(CentOS 4)	Intel(R) Xeon(R) @3.6GHz	4.0GB
Slave node	Linux 2.6.9-55.0.2. Elsmp(CentOS 4)	Quad-Core Intel(R) Xeon(R) @1.60GHz	2.0GB

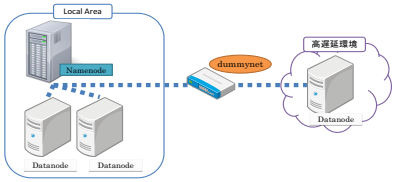

図 1: システム構成

4　高遅延環境における遠隔アクセス制御

4.1　高遅延環境における基本性能

3 台の Datanode のうち 1 台が高遅延環境に存在する場合の基本的な処理性能を計測した．TestDFSIO プログラムでは 10MB のファイルをシーケンシャルライトで 100 個作成，作成したファイルをシーケンシャルリードしファイルシステム I/O 性能を計測した．ここで，“ Throughput ”は，ファイルシステム内部における単位時間辺りのデータ処理量を表している．

ローカルエリアと高遅延マシン間の往復遅延時間 (以下 RTT) を 0msec から 20msec まで変化させ測定を行った．レプリカ数 1 の場合を比較すると，ライトスループットは遅延が増加してもほぼ一定であるのに対し，リードでは大きく低下した（図 2，図 3）．ファイルの書込はバッファを介して行われるためライトでは遅延分の差異が出づらい形となったのに対して，リードではレプリカ数 1 であるため一定の割合で高遅延ノードへのアクセスが行われ，その結果スループットが低下したものと考えられる．

転置インデックスプログラムでは，図 4 より遠隔ノードへの遅延が増加するにつれてプログラム実行時間は増加し，HDFS 内部性能の影響が順当に反映されていることが分かる．

4.2　遠隔アクセス制御時の性能

続いて Hadoop 付属のラック設定を用いて，遠隔ノードへのアクセス制御を行った場合の転置インデックスプログラム実行時間について調べた．まず，遠隔ノードへのアクセスを増加させた場合（simple rack）は最も性能が低い傾向にあることが分かった．またデフォルトの場合（default）と遠隔ノードへのアクセスを制御した場合（optimized rack）を比較すると数回の逆転はあるものの，遠隔ノードへのアクセスを制御した場合の方が実行時間が短い傾向が見られた．今回の実験では個々の処理時間は 700〜800 秒前後であったが，より大量のデータを処理する長期的なジョブを検討した場合，この性能差はますます拡大されるものと予想される．

5　明示的アクセス制御手法の提案

前述の測定では遠隔アクセス制御に Hadoop 付属のラック設定を使用したが，この場合は詳細なアクセス

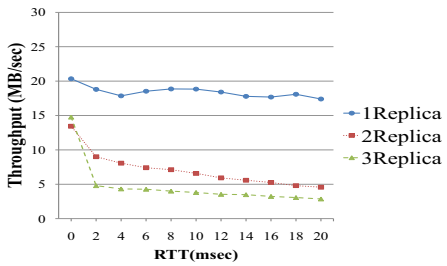

図 2: Write スループット

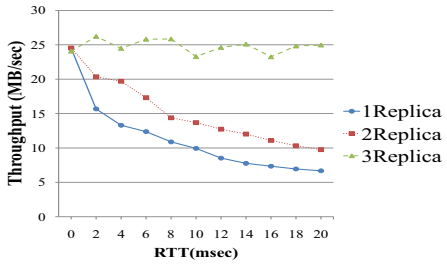

図 3: Read スループット

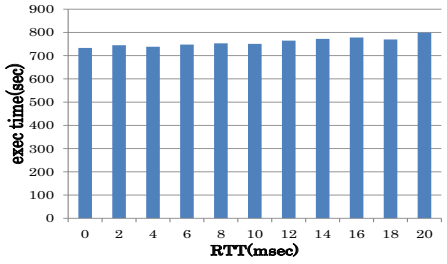

図 4: 転置インデックス実行時間

制御を与えることができない点が課題であった．そこで以下ではファイルシステムを書き換え，直接明示的なアクセス制御を行う手法の検討を行う．またこの手法を用いて高遅延環境における遠隔アクセス制御がシステムに及ぼす影響について，より詳細に検討する．

6 変更後の HDFS

Hadoop では各ノードのネットワークトポロジはラックという単位で認識される．この設定を利用し，(i)1st レプリカはジョブが現在存在するラックへ配置，(ii)2nd レプリカは HDFS 内の Math.random 関数で定義した確率で Remote rack へ，残りを Local rack へ配置，というデータ配置ポリシーに基づいて HDFS を実装した．この際のブロック配置は表 2）のようになり，Remote rack へのデータ配置を最大で 30%軽減する．この値は，Hadoop の耐故障性維持のため必要最低限のデータ分散であるものとした．このようなデータ配置制御を与えた場合の HDFS 性能について，ベンチマークプログラムによる測定を行う．

表 2: ブロック配置 (Remote rack : Local rack)

Math.random 係数	ブロック比	データ配置
0.5 , 0.5	10 : 10	50% - 50%
0.4 , 0.6	9 : 11	45% - 55%
0.3 , 0.7	8 : 12	40% - 60%
0.2 , 0.8	7 : 13	35% - 65%
0.1 , 0.9	6 : 14	30% - 70%

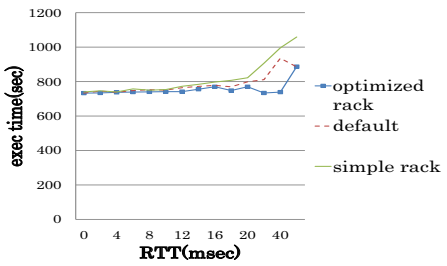

図 5: ラック設定適用時の転置インデックス実行時間

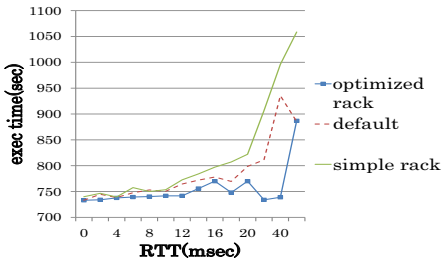

図 6: ラック設定適用時拡大

実際に上記のように変更した HDFS において，一定量のデータ書き込みを与えた際のブロック配置は表（3）の通りである．このとき Math.Random 係数は 0.8 に指定しており（表 2）で求めた通りの 13:7 の割合でデータ配置制御が行われることが確認された．

表 3: データ書込後のブロック配置 (Remote rack : Local rack)

Rack	Node 比	Blocks
rack0	Datanode1	122
rack0	Datanode2	127
rack1	Datanode3	62
rack1	Datanode4	65

7 おわりに

高遅延環境下で自作ベンチマークによる HDFS の性能測定を行ったところ，高遅延環境での性能劣化が確認された．また，遠隔アクセス制御を与え性能測定を行ったところ，遅延ノードへのアクセスが性能へ与える影響が確認された．また明示的なアクセス制御を与える手法として，ラック毎にブロック配置確率を変更する方法を提案し，指定通りのデータ配置制御を実現した．

参考文献

[1] 百瀬 明日香，小口 正人：「高遅延環境における分散ファイルシステム Hadoop の遠隔データアクセス特性の評価」電子情報通信学会　DE 研&PRMU 研 (パターン認識・メディア理解研) 共催 6 月第一種研究会，信学技報，Vol.111，No.76，pp.19-24，2011 年 6 月．

[2] Dhruba Borthakur, *HDFS Architecture*, The Apache Software Foundation. 2008

[3] Rocks Cluster：http://www.rocksclusters.org/