Hadoopにおけるジョブトラッカの耐故障機能の検討

黒松信行† 置田真生† 萩原兼一†

1. はじめに

クラスタ環境においてMapReduceプログラム[1]を実行するためのマスタ・ワーカ型フレームワークとしてHadoop[2]が注目を集めている．Hadoopではマスタであるジョブトラッカ（以降，JT）がタスクをスケジューリングし，ワーカであるタスクトラッカ（以降，TT）にタスクを割当てることでジョブを実行する．Hadoopはジョブの実行に数十から数千のTTを用いる．したがって，TTの耐故障機能が必要となる．Hadoopはタスクを冗長化することでTTに耐故障性を備えている．

一方，JTは耐故障性を備えておらず，単一故障点である[2]．ジョブの実行中にJTが故障すると，ユーザはJTを再起動し，ジョブを再実行する必要がある．

本研究の目的は，ジョブの実行中にJTが故障した場合，ユーザに対して透過的にJTを復元しジョブを継続することである．想定する故障の種類は，故障停止のみとする．以降，故障停止を単に故障と表記する．

2. 提案手法

2.1 透過的な耐故障機能

透過的なJTの復元のためには，故障時にジョブを継続する機構および故障検知が必要である．

2.1.1 故障時にジョブを継続する機構

故障時にジョブを継続する機構にはチェックポインティング（以降，CP）およびロールバックを用いる．JTを実行するノードと異なるノード上でバックアッププロセス（以降，BP）を起動する．JTは，タスク実行が一定数終了する毎にスナップショット（以降，SS）を作成し，BPに送信する．SSはJobTrackerクラスの全フィールドを持つ．BPはJTの故障通知（後述）を受信すると，SSを基にJTを復元（ロールバック）しジョブを継続する．

ただし，CPを用いた復元には情報消失の可能性がある．最後のSS作成から障害発生までの処理に関する情報は失われる．その結果，タスクの重複割当および永遠に終了しないタスクが発生する．

しかし，Hadoopは重複割当を許容する機能を備える．また，長時間終了しないタスク実行を破棄し，再度割当てる機能を備える．したがって，情報消失が発生しても問題にならない．

2.1.2 故障の自動検知

故障検知の機構には，タイムアウトを用いる．Hadoopにおいて，TTはタスクの進捗報告としてハートビート[2]（以降，HB）を定期的にJTへ送信している．そこで，TTはHBの送信に一定回数失敗するとJTが故障したと判断し，BPに故障通知を送信する．JTの復元まで待機した後，TTはHBの送信先を新しいJTに変更する．

ただし，この故障検知には誤検知の可能性がある．例えば，ネットワークの一時的な障害が発生するとTTはHBを送信できない．この場合，JTが故障していないにも関わらず，一部のTTはBPに故障通知を送信し，HBの送信先を新しいJTに変更する．その後直ちにネットワークが復旧すると，誤検知を行ったTTと他のTTではHBの送信先が一致しない．2つのJTが独立にタスクを割当てる状態になるため，タスクの実行効率が低下する．

この問題を解決するため，BPは新しいJTを起動する前に元のJTに停止要求を送信する．これにより，誤検知が発生しても数秒後にJTの数は1つとなる．

2.2 オーバヘッドの見積

提案手法によるオーバヘッドはCPに伴うものおよび復元に伴うものの2つである．

2.2.1 CPに伴うオーバヘッド

CPに伴うオーバヘッドの原因は，SS送信の際TTにおいてタスクの割当待ちが発生するためである．

最悪時のCPに伴うオーバヘッドT_cを予測する．タスク実行が終了すると，TTはHBを用いてJTに終了通知を送信し，次のタスクの割当を要求する．この際，SS作成の条件を満たすと，JTにおいてSS作成のために排他制御が必要となる．T_cが最悪となる

† 大阪大学 大学院情報科学研究科
The Graduate School of Information Science and Technology, Osaka University

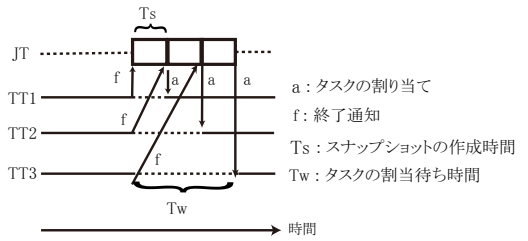

図 1 最悪のオーバヘッドが発生する実行状況（$J = 1$，$P = 3$）

場合は，JT に終了通知の到着が集中し，排他制御が連続する場合である．最悪時の実行状況の例を**図 1** に示す．SS の作成時間を T_s，TT の数を P，SS の作成条件をタスク実行が J 個終了する毎とすると，タスクの割当待ち T_w の最大値は $T_w = T_s * (P/J)$ と表せる．さらに，ジョブを構成するタスク数を N とすると 1 つの TT に対するタスクの割当は約 N/P 回発生する．よって $T_c = T_w * (N/P) = T_s * (N/J)$ と予測できる．

2.2.2 復元に伴うオーバヘッド

復元に伴うオーバヘッドの原因は，故障から復元までの間にタスクの割当待ちが発生するためである．この時間 T_r の最大値は，故障検知に要する時間と JT の復元に要する時間の和となる．これらの時間は P，N，J によって変化するが，その変化量はジョブの実行時間と比較して十分小さい．そのため，T_r はほぼ一定とみなせる．

3. 評 価 実 験

故障時に JT が透過的に復元することを確認する．さらに，提案手法に伴うオーバヘッドを評価する．実験には Hadoop のバージョン 0.20.2 を用い，ギガビットイーサネットで接続された 16 台のクラスタ上で実行した．各ノードマシンのスペックは OS が CentOS 5.2，CPU が Intel Xeon 3.4 GHz Dual，メモリが 2 GB である．

3.1 透過的な復元

故障時の状況として，次の 3 つを想定する．

(1) 復元時に情報消失が発生しない
(2) 情報消失が発生する
(3) 故障を誤検知する

JT の故障はジョブの実行中に JT を kill コマンドにより強制終了することで再現した．

まず（1）の場合，BP が JT を復元し，ジョブを継続できた．その後ジョブは正常に完了し，得られた結果は正しかった．次に（2）の場合，JT は未割当のタスクの報告を受けると，タスクを破棄し新たにタスクを割当てた．また，10 分（Hadoop における設定で変更可能）経過しても終了しないタスクが存在した場合，JT はそのタスクを他の TT に割当てた．これらの結果，ジョブが完了し正しい結果を得られた．最後に（3）の場合，全ての TT が新しい JT に HB の送信先を切り替えてジョブを継続し，正常に完了できた．

表 1 オーバヘッド T_c の予測値と実測値の比較（$N = 163$, $P = 16$）

J	6	12	18
全体の実行時間（s）	515.7	505.3	498.6
T_c の予測値（s）	48.9 (9.4%)	24.5 (4.8%)	16.3 (3.2%)
T_c の実測値（s）	24.1 (4.6%)	13.7 (2.7%)	7.0 (1.4%)

括弧内は全体の実行時間に占める T_c の割合

3.2 オーバヘッドの評価

J を変化させながら Hadoop におけるベンチマークの 1 つである WordCount プログラムを実行した．入力には 22 種類の単語を繰り返し並べた 10 GB のテキストファイルを用いた（$N = 163$）．T_s の実測値は 1.8 秒であった．また，T_r の実測値は 19 秒であった．**表 1** にオーバヘッド T_c の予測値と実測値を示す．表 1 が示すように，T_c の実測値は予測値と比較して約半分であった．

この原因は，実際には各タスクの終了通知が分散するためである．分散の増大に従って T_w は減少する．1 回のタスク割当待ちによるオーバヘッドが減少したため，全体のオーバヘッド T_c が減少した．

4. ま と め

Hadoop において単一故障点である JT に対して CP を用いて耐故障性を向上した．JT の故障検知には HB によるタイムアウトを用いた．提案手法に伴うオーバヘッドは最大 4.6%であった．

謝辞 本研究の一部は科学研究費補助金（基盤研究 (B)2330007）の支援を受けた．また，有益な助言を頂いた大阪大学の土屋 達弘 准教授に感謝します．

参 考 文 献

1) Dean, J. and Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters, *Communication of the ACM*, Vol. 51, No. 1, pp. 107–113 (2008).
2) White, T.: *Hadoop*, オライリー・ジャパン (2010).