3G-04

DMTCPによるノード構成の動的変更に対応した並列分散処理環境の検討

杉山 裕紀 † 澤田 祐樹 †† 大津 金光 †† 横田 隆史 †† 大川 猛 ††
† 宇都宮大学工学部情報工学科 †† 宇都宮大学大学院工学研究科情報システム科学専攻

1 はじめに

近年，モバイル端末の高性能化が進み，新しい並列分散アプリケーションの基盤として注目されている．我々はAndroid端末を利用して，並列分散アプリケーションを実行するAndroidクラスタコンピュータシステムを開発している[1]．本クラスタコンピュータは持ち運びが容易なモバイル端末をノードコンピュータとして利用しているため．並列分散アプリケーション実行中に一部のノードが脱退した場合，処理を途中で中断しなければならない．このため，本システムは並列分散アプリケーション実行中にチェックポイントデータを取得し，これを用いて処理を再開できるチェックポイント/リスタート機能を実装している．また，リスタート時において、最適な負荷分散を行うために、クラスタ内でのノード構成を把握し、その構成に応じて最適な配置でアプリケーションの実行をリスタートするノード構成変更機構を備えている。しかし，現時点でノード構成の動的変更機能は，一部手動で行われており，その完全自動化に必要となる，リスタート処理の開始条件等が未決定のままである．そこで，本機能の完全自動化を実現するための検討を行う．

2 Androidクラスタシステム

本クラスタシステムでは，ノード間の通信には高速かつ容易に利用可能なWi-Fiを用いている．また，並列分散処理のフレームワークとしてOpen MPIを用いている．本システムは，移動体であるAndroid端末をノードコンピュータと用いるため，クラスタを構成する一部のノードコンピュータの脱退や，新たなノードの参入が起こりえる．MPIアプリケーションの実行開始には，並列処理に使用するノードの情報を記述したマシンファイルが必要である．MPIアプリケーションを起動するノード（ホストノード）がマシンファイルを参照することで，ホストノード以外のノード（リモートノード）にプロセスを投入する．しかし，本クラスタシステムではノードの構成が動的に変化するため，並列処理に使用可能なノードをあらかじめ把握することが難しいため，正確に把握する必要がある．

MPIアプリケーション実行中にノードが脱退した場合，脱退したノードと通信ができなくなるため実行中のアプリケーションを継続することは難しい．そのため，本クラスタコンピュータには，MPIアプリケーションの処理を途中から再開する機能が実装されている[2]．

この機能は並列実行途中にチェックポインティングを行うことで実現されている．チェックポインティングにDMTCP[3]を用いている．DMTCPの管理下でアプリケーションを起動するためにdmtcp_launchコマンドが用意されている．これを使ってアプリケーションを起動すると，ホストノードではDMTCPの管理プロセスであるdmtcp_coordinatorが起動する．また，起動したアプリケーションの実行プロセス内には，アプリケーションの実行スレッドとDMTCPのチェックポイントスレッド（CT）が生成される．dmtcp_coordinatorがCTにチェックポイントデータ（以降ckptデータと呼ぶ）を取得する要求メッセージを送信することで，チェックポインティングを開始する．また，ユーザがdmtcp_coordinatorに指示を送るために，dmtcp_commandコマンドが用意されている．これ利用することで，チェックポインティング等を行える．取得したckptデータにはプロセスを復元するために必要な情報があり，取得時に生成されたヘルパスクリプトを実行することで，ckptデータからプロセスを復元しアプリケーションをリスタートすることができる．

現在のAndroidクラスタシステムは，ノード構成の変化によるアプリケーションの中断を自動的に検知して，自動的にckptデータから処理をリスタートすることができない．そこで，DMTCPを利用してノード構成の変更を検知して，自動的にアプリケーションをリスタートする機能を検討する．

3 ノード構成の動的変更への対応

MPIアプリケーションの実行開始時には，ノード構成を把握してマシンファイルを作成する必要がある．また，アプリケーションのリスタート時においても，ckptデータからアプリケーションをリスタートするためには，変更後のノード構成を把握する必要がある．これを実現する方法として並列実行アプリケーション全体を管理する存在であるdmtcp_coordinatorが動的なノード構成の把握と，リスタート処理の制御を行う方法が考えられる．しかし，dmtcp_coordinatorは管理下にあるプロセスが全て終了した後に自身も終了する．そのため，dmtcp_coordinatorが停止している間にも，クラスタ全体を制御するプロセスが存在している必要がある．そこで，dmtcp_coordinatorとは別にクラスタ全体を制御するためのプロセスを立ち上げ，

Consideration on Parallel and Distributed Processing Environment Allowing Dynamic Change of Node Configuration with DMTCP

†Hiroki Sugiyama, ††Yuki Sawada, ††Kanemitsu Ootsu, ††Takashi Yokota and ††Takeshi Ohkawa
Department of Information Science, Faculty of Engineering, Utsunomiya University (†)
Department of Information Systems Science, Graduate School of Engineering, Utsunomiya University (††)

DMTCP 管理下での MPI アプリケーションの実行開始，ノード構成の動的な把握とリスタート処理の制御を行わせる．

MPI アプリケーション実行中にノード構成が変化した場合，実行中のアプリケーションを一旦終了して，クラスタ制御プロセスにリスタートの依頼を行う．dmtcp_coordinator は管理下にあるプロセスの CT と定期的にメッセージを交換し，応答の有無を確認することでノードの脱退を検知し，リスタートを開始できるようにする．また，アプリケーション実行中に参入ノードが発生した場合でも，クラスタ制御プロセスにリスタートの依頼を行う．dmtcp_command コマンドは dmtcp_coordinator に対して様々ユーザコマンドを送信することができる．そこで，dmtcp_command コマンドの機能を拡張して，dmtcp_coordinator に対してノードの参入を通知するユーザコマンドを実装することでノードの参入を検知し，リスタートを開始できるようにする．

アプリケーションのリスタートには ckpt データが必要であるため，dmtcp_coordinator が定期的にチェックポイントの取得要求を行い ckpt データを取得する．プロセスを他のノードに移譲するには，移譲したいプロセスの ckpt データが必要である．ckpt データはノード毎に保存されるため，ノードが脱退した場合そのノードの ckpt データを利用できなくなるため，あらかじめクラスタ内のノードに全ての ckpt データを保存しておく必要がある．また，リスタート処理はホストノードのクラスタ制御プロセスに依頼されるため，CT がチェックポイントデータ作成後にホストノードに送信するようにして，全ての ckpt データをホストノードが保存する．

図 1 に，3 ノードの Android クラスタシステムにおける，ノード脱退時からリスタートまでの流れを示す．MPI アプリケーションを実行中はノードの脱退に備えて dmtcp_coordinator が定期的にチェックポインティングを行う．この時，各ノードで保存された ckpt データは CT によってホストノードに保存される．脱退ノードの検知は，dmtcp_coordinator が管理下にあるプロセスの CT に定期的に生存を確認するメッセージを送信し，応答の有無で判断する．ノード B のように，応答があった場合はアプリケーションの実行を継続する．ノード C のように応答がなかった場合は，ノードが C 脱退したと判断して，管理下にある全てのプロセスをいったん終了する終了する．その後，ホストノードのクラスタ制御プロセスに dmtcp_coordinaor がシグナルを用いてリスタートを依頼する．

dmtcp_coordinator からのリスタート依頼のシグナルをホストノードのクラスタ制御プロセスが受信したら，リスタート処理を開始する．ホストノードのクラスタ制御プロセスは，脱退したノードを特定して脱退ノードのチェックポイントデータを他のノードに送信する必要がある．そのため，ブロードキャスト通信で

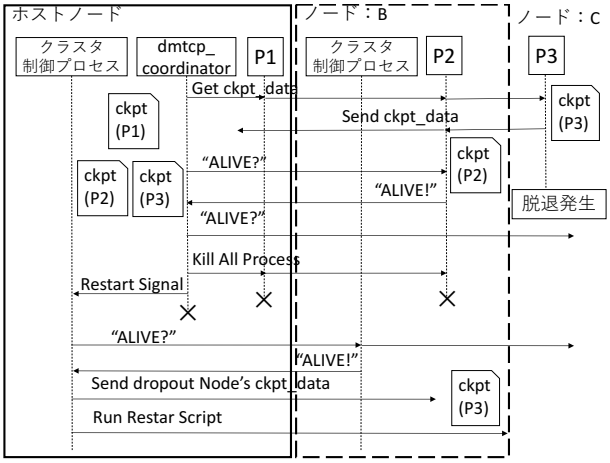

図 1: ノード脱退時における自動リスタート

生存を確認するメッセージを送信し応答を待つ．クラスタ制御プロセスは，ホストノードのクラスタ制御プロセスによって送信された，このメッセージを受信したら応答する．ホストノードのクラスタ制御プロセスは一定時間応答を待ち，応答が返ってきたノード残存しているノードだと判断する．図では，応答はノード B のみであるため，クラスタ制御プロセスはノード C が脱退したと判断する．ホストノードは保存されているノード C の ckpt データをクラスタ内に残っているノード B に送信しヘルパスクリプトを起動する．ヘルパスクリプトを起動すると，脱退したノード C で動作していた P3 がノード B で復元され，アプリケーションがリスタートされる．

4 おわりに

本稿では DMTCP を用いて，ノードの動的変更時に中断したアプリケーションを自動的に再開する機能について検討した．今後の課題として，今回検討した機能を実装して完全な自動リスタートを実現すること挙げられる．

謝辞

本研究は一部 JSPS 科研費 15K00068, 16K00068 の助成による．

参考文献

[1] 荒井 祐介ほか: ”端末の動的な参加・脱退を支援する無線接続型 Android クラスタシステムの実装”, 信学技報, Vol.114, No,155, pp.143-148, 2014.

[2] Y.Sawada,Y.Arai,K.Ootsu,T.Yokota,T.Ohkawa: “An Android Cluster System Capable of Dynamic Node Reconfiguration”, ICUFN2015, pp.689-694，2015.

[3] Jason Ansel, Kapil Arya and Gene Cooperman: “DMTCP: Transparent Checkpointing for Cluster Computations and the Desktop”, 23rd IEEE International Parallel and Distributed Processing Symposium, IPDPS, 2009.