科学研究費助成事業　　研究成果報告書

平成　２６　年　　６　月　　９　日現在

機関番号：　１４４０１
研究種目：　若手研究(B)
研究期間：　2011～2013
課題番号：　２３７０００３６

研究課題名（和文）分散マスタによる高信頼性・高性能ＭａｐＲｅｄｕｃｅの実現

研究課題名（英文）Extended MapReduce framework with distributed masters for high-reliablity and high-performance

研究代表者
置田　真生（Okita, Masao）

大阪大学・情報科学研究科・助教

研究者番号：５０５６３９８８

交付決定額（研究期間全体）：（直接経費）　　3,300,000円、（間接経費）　　990,000円

研究成果の概要（和文）：近年普及が進んでいる大規模並列分散処理フレームワークMapReduceについて、耐故障性の向上およびアプリケーション高速化のための新手法の開発を行った。
まず、耐故障性向上のためにフレームワークを拡張した。従来のフレームワークではマスタと呼ばれる計算機が単一障害点となる。本研究では、複数のマスタを分散配置し障害を自動的に回避する新フレームワークを開発した。
次に、アプリケーションの高速化のために中間データ処理を効率化する手法を開発した。従来手法とは異なる観点から中間データを圧縮することでよりデータ量を削減できること、およびデータの局所性を高めることで中間データの転送量を削減できることを示した。

研究成果の概要（英文）：To improve MapReduce, a popular large-scale distributed programming framework, we developed a high-reliable framework and techniques for accelerating a MapReduce application.
Firstly, we extend an existing MapReduce framework to improving fault-tolerance. In the existing framework, the master computer is the single point of failure. Our extended framework provides multiple master computers and switches them to avoid a system failure.
Secondly, we developed optimizing techniques for intermediate data processing to improve application performance. One of our technique reduces the amount of intermediate data by compressing data from an alternative viewpoint. Another technique reduces intermediate data transfer by improving data locality in task assignment.

研究分野：　総合領域

科研費の分科・細目：　情報学・ソフトウェア

キーワード：　並列処理・分散処理　MapReduce

１．研究開始当初の背景

MapReduce は、Google 社が提唱する分散並列処理向けのフレームワークであり、大規模な PC クラスタ環境で動作する。大規模分散並列ソフトウェアの開発を容易にすることを目的に、分散並列処理の基本的な動作（タスクのスケジューリンク、故障時のリカバリ、ノード間通信など）をフレームワークに隠蔽し、ユーザの負担を軽減する。当時は企業でも続々と採用されるなど、普及が進んでいた。

ただし、当時の MapReduce 実装にはいくつかの問題が存在した。本研究の背景として次の 2 点を挙げる。

（１）耐故障性に関する問題：MapReduce を構成する 2 種類の計算機のうち、ワーカの故障に対してはソフトウェアレベルで対処する様々の手法が実装されていることに対し、マスタの故障については何の対処もされていない。

（２）実行性能に関する問題：MapReduce は使いやすさと拡張性を重視する一方で、性能はあまり重視されておらず、非効率的な実装が存在する。結果、使用するハードウェアの性能を十分に引き出せていない。

２．研究の目的

本研究では、MapReduce 実装の 1 つである Hadoop を拡張し、マスタの機能を複数ノードに分散・協調して提供する手法（以下、分散マスタ）を考案する。本研究の目的と具体的な内容を以下に示す。

（１）マスタを多重化することによる耐故障性の向上

①複数マスタ間におけるデータ一貫性を保つための機構の開発：マスタ間で分散スナップショットを作成し、障害発生時に一貫性が損なわれないことを保証する。

②プログラムから障害を隠蔽する機構の開発:少なくとも 1 つのマスタが動作する限り、プログラムの実行を完了できるアルゴリズムを考案し、プログラムの実行時間を保証する。

（２）複数のマスタに処理を分散することによる実行性能の向上

①マスタ数の自動調節：問題規模とワーカ数にあわせて、自動的に最適なマスタ数を決定し、 必要な数のマスタを起動する。

３．研究の方法

本研究の当初の計画では、Hadoop の実装をもとに、次の 2 つの研究を行う予定であった。

（１）Hadoop のマスタを多重化することによる耐故障性の向上

（２）Hadoop で複数のマスタに処理を分散することによる実行性能の向上

研究途中より、（２）の方法による実行性能向上の期待が小さいことから予定を変更し、代わりに以下の方法による性能向上についての研究を行った。

（３）Hadoop アプリケーション実行中のデータの動き（データフロー）の改善

（１）および（３）に関して、それぞれ具体的な方法を以下に示す。

まず（１）においては、Hadoop のマスタが持つ主要な機能を複数のマスタに分散し、協調動作することで、全マスタが故障しない限りプログラムの実行時間を保証できる手法を開発する。具体的には以下のような手法を採る。

- 複数のマスタ間に親子関係は作らず、全て同等とする
- 複数マスタのうち少なくとも 1 つのマスタが故障なく実行完了できると想定する
- タスクリストは実行中に増減しないため、予め全マスタに配布する
- タスクリストを分割し、マスタごとに割当に責任を持つ範囲を決める
- 適宜、マスタ間でタスクの割当情報を同期する
- 自身が責任を持つタスクを全て完了したマスタは、他のマスタが責任を持つタスクのうち、未完了・未割当のタスクを選んでワーカに割り当てる
- この際に、異なるマスタが同じタスクを重複実行することを許す

次に（３）においては、特定のアプリケーションに限らず、汎用的にデータフローを改善し、アプリケーション実行時間を短縮できる手法を開発する。これを以下の 3 段階に分けて段階的に取り組む。

①データフロー分析ツールの作成：アプリケーションごとにデータフローの特徴を把握するため、データフローを可視化し、実行のボトルネックを特定するための支援ツールを作成する。

②データ圧縮によるデータ量の削減：データの重複などの特徴を利用し、アプリケーション実行中に必要に応じてデータを圧縮することで、データの量を削減しデータフローを改善する。

③局所性の向上によるデータフローの改善：可能な限りデータを移動させずに処理できるようなタスクの割り当てを行い、移動のためのデータ転送を削減しデータフローを改善する。

４．研究成果

（１）Hadoop の耐故障性向上に関する主な成果：マスタの故障から自動的に復帰するソフトウェアシステムを実現した。このシステムは、マスタのバックアップデータを定期的に作成する。故障発生時にはそれを自動的に検知し、バックアップデータを元にマスタを復元する。これらの操作は自動的に、かつアプリケーションの処理と並行して行われるため、ユーザが故障の有無を意識する必要はない。また元来のシステムと比較して、実行時間の増大は高々2%であり、耐故障性を効率的

に実現している。このシステムの特徴は、ハードウェアの障害をソフトウェアでリカバリーできる点にある。国際会議での発表後、海外研究者からの問い合わせが 2 件あり、同様の耐故障性向上に取り組む研究者から関心を得ている。今後の展望として、Hadoop を実装・公開している Apache プロジェクトでも Hadoop マスタの耐故障性機能の開発が進んでおり、それらと連携のうえ収斂していくものと考えている。

（２）データフローの改善による Hadoop アプリケーション高速化について

①データフロー分析ツールに関する主な成果：アプリケーション実行時にデータフローの改善に必要な情報を収集し、可視化するツールを作成した。このツールはアプリケーションの性能ボトルネックおよびその性能を向上させるためのパラメータを自動的に検出し、直観的な図形式で表示する。このツールを利用することで、Hadoop に十分習熟していないユーザであっても少ない労力で Hadoop アプリケーションの性能を改善できる。

②データ圧縮によるデータ量の削減に関する主な成果：Hadoop における代表的なアプリケーションの１つである PageRank について、データフローの重複を削除し高速化する手法を開発した。この手法は既存の高速化手法と比較して最大 1.57 倍の高速化を達成した。この手法の意義は、広く知られた既存手法と直交する観点における重複を利用する点にあり、既存手法があまり有効でないアプリケーションに対する高速化を期待できる点にある。この成果を査読付き国内会議（該当分野では国内最大規模）に投稿し、優秀若手研究賞を受賞した。今後の展望として、この手法を一般的な Hadoop アプリケーションに適用できるように拡張した汎用的手法を開発し、アプリケーションの特徴に応じて自動的に圧縮方法を切り換える手法を MapReduce フレームワークに組み込むことを計画している。

③局所性の向上によるデータフローの改善に関する主な成果：MapReduce アプリケーションを構成する 2 種類のタスク（Map および Reduce）のうち、Reduce におけるデータの局所性を高めるようなタスクの動的割り当て手法のプロトタイプを開発した。割り当ての決定にはアプリケーション実行中における中間データ発生の統計的な予測を用いている。一般的な静的割り当て手法と比較して、MapReduce ベンチマークアプリケーションに対して最大 16%の高速化を達成した。まだプロトタイプの段階であり、適用範囲が狭い。実用化および汎用化のためには実験と改善が必要である。

５．主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計３件）

①. Tomohiro Okuyama, Masao Okita, Takeshi Abe, Yoshiyuki Asai, Hiroaki Kitano, Taishin Nomura, and Kenichi Hagihara. ``Accelerating ODE-based Simulation of General and Heterogeneous Biophysical Models using a GPU''. IEEE Transactions on Parallel and Distributed Systems, 査読有, （採録決定）.http://dx.doi.org/10.1109/TPDS.2013.198

②. Yoshiyuki Asai, Takeshi Abe, Hideki Oka, Masao Okita, Kenichi Hagihara, Samik Ghosh, Yukiko Matsuoka, Yoshihisa Kurachi, Taishin Nomura, and Hiroaki Kitano. ``A Versatile Platform for Multilevel Modeling of Physiological Systems: SBML-PHML Hybrid Modeling and Simulation''. Advanced Biomedical Engineering,査読有, Vol. 3, pp. 50--58, (2014). http://dx.doi.org/10.14326/abe.3.50

〔学会発表〕（計２２件）

①. Yoshiyuki Asai, Takeshi Abe, Hideki Oka, Masao Okita, Tomohiro Okuyama, Kenichi Hagihara, Samik Ghosh, Yukiko Matsuoka, and Hiroaki Kitano. ``A versatile platform for multilevel modeling of physiological systems: Template/instance framework for large-scale modeling and simulation''. In 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'13),Osaka, Japan, (2013 年 7 月 6 日).

②. 黒松信行, 置田真生, 萩原兼一. ``MapReduce を用いたグラフアプリケーションにおける重複メッセージの排除による高速化''. 第 11 回先進的計算基盤システムシンポジウム論文集（SACSIS 2013）, 仙台, 日本, (2013 年 5 月 22 日). 優秀若手研究賞

③. Nobuyuki Kuromatsu, Masao Okita, and Kenichi Hagihara. ``Evolving fault-tolerance in Hadoop with robust auto-recovering JobTracker''. In Proceedings of the 3rd International Conference on Networking and Computing (ICNC 2012), Okinawa, Japan, (2012 年 12 月 7 日).

④. 古谷達朗, 置田真生, 萩原兼一. ``Hadoop の性能ボトルネックを特定するための実行トレース可視化ツール''. 平成 24 年度情報処理学会関西支部支部大会講演論文集 , 大阪, 日本, (2012

年 9 月 21 日).
⑤. Yoshiyuki Asai, Takeshi Abe, Masao Okita, Tomohiro Okuyama, Nobukazu Yoshioka, Shigetoshi Yokoyama, Masaru Nagaku, Kenichi Hagihara, and Hiroaki Kitano. ``Multilevel modeling of Physiological Systems and Simulation Platform: PhysioDesigner, Flint and Flint K3 service''. In Proceedings of the 12th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT 2012), Izmir, Turkey, (2012 年 7 月 17 日).
⑥. 井上佑希, 置田真生, 萩原兼一. ``系列パターン抽出の MapReduce 実装におけるタスク分割方式の検討''. 情報処理学会研究報告, 2010-HPC-130,鹿児島, 日本, (2011 年 7 月 29 日).
⑦. 黒松信行, 置田真生, 萩原兼一. ``Hadoop におけるジョブトラッカの耐故障機能の検討''. 第 9 回先進的計算基盤システムシンポジウム論文集 (SACSIS 2011), 東京, 日本, (2011 年 5 月 26 日).

６．研究組織

(1)研究代表者

置田 真生（OKITA, Masao）
大阪大学・大学院情報科学研究科・助教
研究者番号：５０５６３９８８