

審査の結果の要旨

氏 名 陳 婷

本論文は, **ParaLite: a Parallel Database System for Data-intensive Workflows** (邦題: **ParaLite**: データ集約的ワークフローのための並列データベースシステム)と題し, 各応用分野の研究者がワークフローアプリケーションを容易に記述し, 効率よく実行するためのシステムについて述べたもので, 全体で9つの章からなる. 本文は英文で書かれている.

第一章は, **Introduction** (序論)と題し, データ集約的な計算の重要性, そのような計算がしばしば既存のプログラムを大量データに対して並列に適用し, 複数の既存プログラムを組み合わせることで構成されることについて述べている. さらに, 既存のアプローチとして, ファイルに基づくワークフローシステム, 近年しばしば用いられる **MapReduce** に基づくシステム, 関係データベースに基づくシステム, それぞれの問題点を述べている. そして本研究で提案するアプローチが, 関係データベースシステムを起点としつつ, 問い合わせに既存プログラムを容易に統合しそれを効率的に並列実行できるよう拡張したものであることを述べている.

第二章は, **Background** (背景)と題し, 大量データ処理をサポートするシステムの設計法として関係データベースに基づくもの, **MapReduce** に基づくものそれぞれについて延べ, それらの利害特質やそれぞれの問題点について述べている. 特に関係データベースを設計上の起点とすることの妥当性, 一方で既存の関係データベースが抱える限界(既存の実行可能プログラム統合の難しさ, それを行った場合に生ずる性能の問題, 長時間実行される問い合わせに対する耐障害性のなさ)について述べている.

第三章は, **Related Work** (関連研究)と題し, 本研究と目標を共有する個別の研究との関連性について論じている. **MapReduce** に **SQL** 風の拡張を施したいくつかのシステム, **MapReduce** とデータベースを組み合わせたシステム, データベースを **MapReduce** 風の機能で拡張したシステム, 既存のファイルベースのワークフローシステムなどについて述べている. また, 耐故障性に関して **MapReduce** に基づくシステムが細粒度な耐障害性を提供していること, 関係データベースの文脈では, 問い合わせ実行途中の障害発生を効率的に扱う研究が最近になって表れていることについて述べている.

第四章は, **System Design** (システム設計)と題し, 提案するシステム **ParaLite** の設計について述べている. システムを構成するソフトウェア部品とその間の相互作用, データモデル, 問い合わせから実行プランの生成, 実行の方式などについて述べている. また, 通常の関係データベースシステムと異なりユーザがサーバを管理する必要がないなど, 応用分野の研究者が容易に使えるようにするための, 設計上の特徴について述べている. そして,

ParaLite を用いてワークフローがどのように記述されるのかについていくつかの典型パターンを用いて説明している。

第五章は, **Integration of Executable** (実行可能プログラムの統合)と題し, ParaLite において既存プログラムをどのように問い合わせに統合するのかを説明している。SQL を元にした記法上の拡張である **User Defined Executable (UDX)**, UDX の実行モデル, 並列実行(負荷分散)のアルゴリズムについて述べている。また, 関係データベースが備える類似の機能として見られる **User Defined Function (UDF)**との違いについて述べ, 拡張の必要性について述べている。

第六章は, **Intra-query Fault Tolerance** (クエリ内耐故障)と題し, SQL 風の文法で長時間実行されるワークフローの記述を行うシステムが直面する耐故障性の問題について述べている。SQL の問い合わせを, 実行プランを表す DAG として表現し, 「DAG 内のどのノードの出力を永続記憶に保存すべきか」という最適化問題として定式化している。また, そのモデル化のもとで計算量が少ない, 経験的な最適化アルゴリズムを提案している。

第七章は, **Evaluation** (評価)と題し, システムの基本性能を評価している。評価対象として MapReduce 処理系である Hadoop, それに SQL 風の拡張を施した Hive, 商用並列データベース DBMS-X との比較を行なっている。ワークロードとして選択, 集約, 結合などの単純な問い合わせ, TPC-H ベンチマークを評価している。次に, 実行可能なプログラムを UDX を用いて問い合わせ中に統合することで, 通常の UDF のみをサポートする既存データベースシステムを, 性能で大きく上回ることを示している。さらに, 負荷分散の方式, 耐故障方式についても, 基本性能の評価を行なっている。

第八章は, **Real-world Text Processing Workflows** (実世界テキスト処理ワークフロー)は, 実応用として, 3 つのテキスト処理ワークフローを ParaLite やその他のシステムを用いて記述・実行した際の知見について述べている。アプリケーションは, 生物医学文献の深い解析に基づくイベント抽出タスク, 日本語の形態素解析を用いた単語数カウント, および統計的アプローチを利用した文のチャンキングの 3 種類である。いずれも自然言語処理のモジュールを利用した, 一から書くことは現実的でないアプリケーションであり, 本性の評価によって, ParaLite がそのようなアプリケーションを, 既存のシステムと比べて容易に構築できることが実証されている。

第九章は, **Conclusions** (結論)と題し, これまでの知見がまとめられている。

付録第 A 章では, **Workflows Description** (ワークフローの記述)と題し, 第八章の評価で用いられたアプリケーションの記述が完全な形で掲載されている。

以上これを要するに本論文は, データベース言語である SQL を設計上の起点として, 既存の実行可能プログラムを組み合わせたワークフローを容易に記述し, 効率的に並列実行できるシステムの設計と実装方法を論じ, ベンチマークおよび実際の応用を通してその有効性を実証したもので, 電子情報学上貢献するところが少なくない。よって本論文は博士 (情報理工学) の学位請求論文として合格と認められる。