

論文の内容の要旨

論文題目	言語処理系におけるごみ集めとコード最適化に関する研究
学 位 申 請 者	鈴木 貢

本研究では、言語処理系の実装に関連する二つの事項を取り扱う。本論文の第1部（第2章～第4章）では、実行時系における記憶場所の動的な管理である「ごみ集め」に関して議論する。ごみ集めは、記号処理言語やオブジェクト指向型言語の実装には不可欠なしくみで、その性能が全体の性能を左右する。そして、第2部（第5章～第8章）では、SIMD (Single Instruction Multiple Data-stream) 命令の一種のレジスタ分割型ショートベクタ命令セット（以下SIMD命令と略す）に注目し、翻訳系によるSIMD命令向け最適化に関して議論する。この命令セットは、メディアデータ処理の高速化の安価な手法として考案され定着してきた。これには、従来のコード最適化技法を応用可能な事項もあるが、新しく開発する必要がある事項も多い。現状では、コンパイラによるSIMD命令の自動生成は研究段階であるので、アセンブリ言語等を用いないとSIMD命令の満足がいく活用ができない。

第1章では、二つの事項を研究するに至る背景について議論する。

第2章では、先ず、第1部で取り扱うごみ集め問題を定式化した。次に、参照計数式、複写式、印掃式というごみ集めの三つの戦略を示した。そして、従来の印掃式ごみ集めでは解決されていなかった二つの問題を指摘し、関連する既存の成果を紹介し分析した。

第3章で議論している第一の問題は、印掃式ごみ集めの実行時間を小さくすることである。従来の印掃式ごみ集めは、記憶領域のサイズを S として、 S に比例する時間が必要であった。これに対し、使用中の記憶の量を A として、理論上の計算量は $A \log A$ 、実用上は A に比例する実行時間で済むごみ集めで、印掃式かつ使用領域を圧縮するもの（以下印掃式圧縮型と略す）を提案した。これにより、 A が S の1割以下という一般的な状況の下では、ごみ集めが高速化された。

第4章で議論している第二の問題は、「世代別ごみ集め」と呼ばれるごみ集めの手間を省く戦略を、印掃式圧縮型ごみ集めに適用することである。世代別ごみ集めは、通常は、複写式のごみ集めとの組み合わせで実現される。ここでは、印掃式圧縮型ごみ集めの生成順序保存という特徴を生かした方法を提案し、小さなオーバーヘッドで世代別ごみ集めを実装する方法を示した。これにより、さらにごみ集めが高速化された。

第5章では、先ず、第2部で取り扱うSIMD命令の動作や種類を説明する。次に、コンパイラによるSIMD命令の自動生成に関する問題点を指摘し、関連する既存の成果を紹介し分析す

る。そして、本研究での実装に関連する成果を紹介する。

第6章で議論している第一の問題は、SIMD命令向けの言語仕様の拡張を使用していない(つまり通常の言語規格の範囲内で記述された)プログラムから、SIMD命令の振る舞いに相当するオペレーションを抽出する方法である。このために、覗き穴最適化を応用した方法による最適化技法を提案した。

第7章で議論している第二の問題は、高級言語の汎整数拡張が、効果的なSIMD命令の適用を阻害していることである。汎整数拡張とは、汎整数型(通常は計算機が効率よく処理できる整数型)より小さいサイズのデータは、演算に使用される前に汎整数型への符号拡張やゼロ拡張が施されるという仕様である。これに対して、なるべく小さいデータサイズによる演算で、汎整数拡張を行う場合と同じ結果を得るためのプログラム解析法を提案した。

第8章で議論している第三の問題は、実在する多くのベクタ処理向きのプログラムやベンチマークプログラムが、コンパイラによるSIMD命令活用のベンチマークに適していないことである。そこで、これらの中の例題を分析し、SIMD命令を効果的に適用できるように変形し、SIMD命令向きのコーディングを示した。さらに、コンパイラのSIMD命令の活用度を調べるためのベンチマークとして、各々の例題に対して、SIMD命令の適用を容易にする種々の変形を組織的に施したプログラム集で、複数の変形版から成るものを用意した。

最後に第9章では、本研究で得られた成果を総括する。成果の一つは、記憶領域の使用効率が高く、実行時間も小さいごみ集めを提案したことである。それにより、記憶領域が乏しい計算機でも、近代的な言語が必須とするごみ集めを効率よく実現できる。もう一つは、SIMD命令向けの最適化技術を提案し、実装し、評価し、実行性能の向上を示したことである。これにより、命令セットが異なる計算機間でも、同じメディアデータ処理プログラムを共有できるようになる。

論文審査の結果の要旨

学位申請者氏名	鈴木 貢
審査委員主査	野下 浩平
委員	尾内 理紀夫
委員	小林 聡
委員	岩田 茂樹
委員	岩崎 英哉
委員	中山 泰一
委員	渡邊 坦

本論文は、言語処理系に関する2つの研究課題をとりあげている。

まず、本論文の概要を説明する。第1の課題は、実行時系における動的な記憶管理法であるごみ集め (garbage collection) である。ごみ集めは、記号処理言語やオブジェクト指向型言語の処理系に必須の構成要素であり、その性能が処理系全体の性能を左右する。本論文では、印掃式ごみ集めで使用記憶領域を圧縮する方式を対象にして、実行効率のよい解決法を示した。第2の課題は、SIMD (Single Instruction Multiple Data-stream) 拡張命令に対する翻訳系による目的コードの最適化である。メディアデータ処理ではレジスタ分割型ショートベクタ命令という SIMD 命令群が使われている。この型の命令を活用する方法は、多くの場合低水準言語の使用等によるものであり、自動生成の研究は十分に進んでいない。本論文では、命令自動生成に関する3つの研究課題、すなわち SIMD 命令列の最適化、整数の型に関するプログラム解析、SIMD 命令用ベンチマークの作成、それぞれに対する解決法を提示した。

次に、本論文の構成に従って本研究の成果と審査の結果を述べる。

第1章では、研究課題の背景について議論している。

第2章では、まずごみ集めの問題を定式化し、次に参照計数式、複写式、印掃式という3つの戦略を説明している。そして、従来の印掃式ごみ集めで解決されていなかった問題点を指摘し、関連する従来の研究を概観している。

第3章の課題は、印掃式ごみ集めの能率の向上である。従来の印掃式ごみ集めは、記憶領域の大きさを S とすると、 S に比例する時間が必要であった。これに対して、使用中の部分の大きさを A とすると、理論上の計算量が $A \log A$ 、実際上は A に比例する実行時間ですむごみ集めの方法で、印掃式圧縮型 (印掃式かつ使用領域を圧縮する型) のものを考案した。この方法によって、 A が S の1割以下という実際的な状況でごみ集めが高速化できることが示された。この結果は、ごみ集めのアルゴリズムとして優れたものであると評価できる。

第4章の課題は、ごみ集めの能率を向上するための戦略の中で代表的な世代別ごみ集めを印掃式圧縮型ごみ集めとして実現することである。世代別ごみ集めは、通常複写式のごみ集めとの組み合わせで実現される。本論文では、印掃式圧縮型ごみ集めの生成順序保存という特徴を生かした方法を提案し、小さなオーバーヘッドで世代別ごみ集めを実装する方法を示した。こ

の方法によって本論文のごみ集めの方法がさらに高速化された。この結果はごみ集めの現代的なアルゴリズムとして優れたものであると評価できる。

第5章では、まず本論文の後半でとりあつかう SIMD 命令の動作や種類を説明している。次に、コンパイラによる SIMD 命令の自動生成に関する問題点を指摘し、これに関連する従来の研究を概観している。そして本研究での実装に関する成果を述べている。本論文の新しい研究結果は、第6～8章に詳しく説明されているが、これらは現在の研究動向に沿うものであり専門的に優れたものであると評価できる。

第6章の課題は、SIMD 命令向けの言語仕様の拡張を使用していないプログラムから、SIMD 命令の振る舞いに相当する操作を抽出する方法の開発である。この実現のために、覗き穴最適化法を応用した新しい最適化技法を提案している。

第7章の課題は、高級言語の汎整数拡張が効率的な SIMD 命令の適用を阻害していることへの対応策の提示である。汎整数拡張とは、計算機が効率よく処理できる整数型より小さいサイズのデータに対して、それが演算に使用される前に通常の整数型への符号拡張やゼロ拡張が施されることである。本論文では、なるべく小さいデータサイズによる演算によって、汎整数拡張を行う場合と同じ計算結果を得るために、新しいプログラム解析法を提案した。

第8章の課題は、実在するベクタ処理向きのプログラムやベンチマークプログラムがコンパイラによる SIMD 命令活用のベンチマークに適さないことに対する解決策の提示である。本論文では、従来のプログラム集を変形して SIMD 命令向きのコーディングを示すとともに、SIMD 命令の活用度を調べるためのベンチマークのプログラム集で、複数の変形版からなるものを作成した。

最後の第9章では本論文の成果を述べている。まとめると、実行時間と記憶領域に関して能率のよいごみ集めの方法を提示したこと、および SIMD 命令向けの最適化技術に関する新しい結果を示したことである。

以上のとおり、本論文は言語処理系の専門分野において優れた研究結果を示したものであり、博士の学位請求論文として十分な価値を有しており合格と判定する。