

Title	最適化器の生産性向上を目的としたコンパイラフレームワークの設計と実現
Author(s)	斉木, 晃治
Citation	
Issue Date	2003-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1671
Rights	
Description	Supervisor: 権藤 克彦, 情報科学研究科, 修士

最適化器の生産性向上を目的とした コンパイラフレームワークの設計と実現

齊木 晃治 (110046)

北陸先端科学技術大学院大学 情報科学研究科

2003 年 2 月 14 日

キーワード: XML, コンパイラフレームワーク, 中間表現, 最適化器, DTD 設計.

1 背景

マイクロプロセッサがその性能を十分に発揮するためには、高度に最適化されたコードが必要である。このため、さまざまな最適化手法が研究されている。最適化手法の研究にはコンパイラの構築が欠かせないが、これには多大なコストと時間を要する。

この問題を解決するために、コンパイラフレームワークの研究が行なわれている。コンパイラフレームワークは、コンパイラの中で再利用可能なコンポーネントを再利用し、コンパイラ構築のコスト削減と再利用性の向上を図ったものである。現在、多くのコンパイラフレームワークが構築され、運用されている。これらは、全てオブジェクト指向のフレームワークとして実現されている。

既存のコンパイラフレームワークは、すでに実装された最適化器の再利用に適している。これは、コンパイラフレームワークの提供する特定の言語向けのプログラミングインターフェースに基づいて実装されている必要がある。しかし、利用できるインターフェースと実装言語を制限することは、開発手法の選択の幅を狭めることになる。このため、生産性の向上が期待できる新たな開発手法を、最適化器の実装に導入できないのが現状である。

2 目的

本研究の目的は、最適化器の生産性向上を目的とした新たなコンパイラフレームワークを構築することにある。我々は、これをXML(Extensible Markup Language)を用いて実現し、評価検討する。XMLは構造化文書のための標準規格である。XMLにはDOM(Document Object Model), XSLT(Extensible Stylesheet Languages Transformations)などの関連技術が多数存在する。

既存のコンパイラフレームワークはオブジェクト指向の API を提供している．これに対して，本研究で提案するフレームワークは，XML のデータスキーマを中心に構築される．これは，既存のコンパイラフレームワークと対極に位置する．本研究では，XML とその関連技術が最適化器の生産性向上に有効であることを，実装実験によって確認する．

3 XML に基づくコンパイラフレームワークの実現

XML に基づくコンパイラフレームワークは，コンパイラの間中表現を XML でマークアップすることで実現する．コンパイラとしての機能を提供するコンポーネントは，XML 文書と中間表現のマッピングを定義するだけでよい．最適化器は XML 文書に対して最適化を行なう．XML のデータスキーマに従っていれば実装言語は任意に選択可能である．

最適化器の行なう主な処理は，1. 依存関係の取得，2. 解析，3. 中間表現の変更と出力，の 3 つである．XML ベースの最適化器は，XPath を用いて依存関係を取得し，DOM を用いて XML で表現された中間表現を操作する．

本研究では，XML に基づくコンパイラフレームワークの実現を 2 つの方法で行なう．1 つは，既存のコンパイラを拡張する方法で，GCC(GNU Compiler Collection) を拡張する．もう 1 つは，新たにコンパイラの構築から行なう手法の 2 つの手法で，C 言語サブセットの XC 言語を対象にする．前者は，実用的なコンパイラフレームワークの実現が可能だが，GCC のソースコードの解析が必要である．後者は，中間表現をより理想的な形で実現可能だが，コンパイラの構築が必要である．

4 既存のコンパイラをベースとするコンパイラフレームワークの構築

既存のコンパイラをベースにする手法では，GCC の中間表現である RTL(Register Transfer Language) を XML で表現するマークアップ言語 RTL-XML を設計した．RTL-XML を用いることで，プログラミング言語とマシンアーキテクチャの双方に依存しないツールの実装が可能である．この例としては，アスペクト指向言語における weaver の実装などが考えられる．ここでは，保存された RTL-XML 文書を最適化以外の目的に利用する例として，RTL をブラウザ上で表示できる視覚化ツールを試作した．

コンパイラフレームワークの実装は，GCC の RTL デバッグ出力機能を拡張することで行なった．しかし，RTL と GCC の内部データ構造の依存関係に起因する問題から，完全に動作する実装には致らなかった．

この主たる要因は，我々が実装の初期段階において，RTL と GCC の内部データ構造の間の依存関係を完全に把握していなかったことによる．これによって，GCC の拡張性と詳細な技術文書の不足が明かになった．

5 C 言語サブセットに対するコンパイラフレームワークの構築

コンパイラフレームワーク構築のもう1つの手法として、C 言語サブセットの XC 言語を対象に、コンパイラフレームワークの構築した。これは、XC の処理系である `xcc` の設計と実装も含んでいる。これによって、コンパイラの内部データ構造と中間表現を分離し、XML での出力を容易にする設計が可能である。ここでは、XC の中間表現を XML で表現するマークアップ言語 XCC-XML を設計した。また、XCC-XML 処理系の行なう最適化を制御する ODF(Optimizer Description File) の設計も併せて行なった。

実装したコンパイラフレームワーク上で最適化器の実装実験を行なった。実装はオブジェクト指向のスクリプト言語である Python 言語で行なった。4つの最適化手法を実装し、それぞれの最適化器の実装に要した期間は1日前後であった。

C 言語のサブセットを対象としたこの実験では、XML に基づくコンパイラフレームワークの有効性を実証するには不十分である。我々は、これを実用的なコンパイラフレームワーク構築のための予備的実験と考える。

6 議論

XCC-XML を用いた予備実験の範囲で、XML に基づくコンパイラフレームワークの有効性を確認した。その理由を以下に示す。

- XPath は、中間表現の探索に十分な表現能力を持っている。
- 一旦、XCC-XML 文書を生成し、保存することで、デバッグやテストのたびに生成する必要がなく、生産性の向上に効果的である。
- Python インタープリタ上にロードした XCC-XML 文書に対して、さまざまな実験やテストを対話的行なうことができる。実験のたびに読み込む必要はない。

これはあくまでも予備実験の範囲で確認したものである。実用的なコンパイラフレームワークとして、その有効性を実証するものではない。これには、コンパイラフレームワークとしての実用性の向上が必要である。

7 まとめ

本研究では、最適化器の研究開発における生産性向上を目的として、XML を用いたコンパイラフレームワークを提案した。これを実現するために、我々は、GCC をベースにした手法と新規にコンパイラを実装する手法の2種類の実装を試みた。前者の実装は不完全なものに終わり、後者の実装を用いて、最適化器の実装実験を行なった。

実験における実装に限定すれば、XML とその関連技術は有効に機能した。また、XML によって可能になった最適化器の開発手法の有効性も確認した。しかし、XPath による記述の再利用性に関しては、検討の必要が認められた。

後者の実装である XCC-XML とその処理系について、改善の必要性を認識した。現状の XCC-XML のスキーマでは、ローカル変数のレジスタへの割り付けや命令のスケジューリングのような実用的な最適化を行なえない。処理系に関しても、C 言語サブセットではなく、ANSIC や GCC 拡張に対応することが必要である。

当面の課題を解決した上での、今後の課題を次に示す。

- 最適化の種類と目的に応じた多階層の中間表現の設計と実現
- 複数アーキテクチャへの対応
- GCC をはじめとする既存のコンパイラとの互換性の確保
- XML で表現された中間表現の最適化以外の目的での利用法の検討
- XPath を再利用するためのリポジトリの構築