

MPI を用いた ISIS-SimpleScalar の実装

鈴木 幸夫[†]久保田 陽介[†]埜 敏博^{††}

東京工科大学 工学部 情報工学科[†]
東京工科大学 コンピュータサイエンス学部^{††}

1 はじめに

並列計算機を開発する際に、実装前段階でシミュレーションによりその性能を予測することによって、低コストで柔軟性に富んだ開発を行うことができる。

並列計算機シミュレータ ISIS は、C++ によって実装されており、プロセッサやメモリなどの機能ブロックシミュレータを集めたクラスライブラリである。ISIS のプロセッサ・ユニットとして、米 IDT 社の R3081 が用意されている。しかしながら、最新のプロセッサを用いる場合は、複雑な処理のシミュレータを新たに実装する必要がある。そこで、高機能なプロセッサ単体のシミュレータである SimpleScalar を ISIS のプロセッサユニットに利用することができれば、様々なプロセッサを使用した並列計算機シミュレータを比較的容易に作成することができる。

本研究では、SimpleScalar のプロセッサシミュレータと ISIS のプロセッサユニットを別プロセスで実行し、MPI で接続することにより、両シミュレータの違いを吸収し柔軟性の高いシミュレーション環境を実現する。

2 ISIS

ISIS[1] は、慶應大学の若林正樹氏らによって開発された、並列計算機シミュレータ作成のための支援ライブラリである。プロセッサやメモリといった並列計算機内の個々の機能ブロックシミュレータを集めたもので、C++ 言語によって実装されており、これらを目的に併せて選択することで、様々なアーキテクチャの並列計算機シミュレータを構築することができる。機能ブロックシミュレータをユニット、各ユニットを相互接続するため、機能ブロック間の接続やバスやルータ間の結合路などの入出力端子を担うポート、データなど

Implementation of ISIS-SimpleScalar using MPI

[†]SUZUKI Yukio

[†]KUBOTA Yousuke

^{††}HANAWA Toshihiro

[†]Dept. of Information Technology, Tokyo University of Technology

^{††}Dept. of Computer Science, Tokyo University of Technology

の送受信される情報を表すパケットの3つの概念を用いてシミュレートする計算機をモデル化する。ユニットは外部からのクロック入力により、接続されたユニット以外から独立して動作し、シミュレータ内の全ユニットの動作順序を保証するために、ユニットのクロック入力を入力フェーズを出力フェーズに分割している。

3 SimpleScalar

SimpleScalar[2] は、SimpleScalarLLC によって開発・保守が行われている。C 言語によって実装されたプロセッサ単体のシミュレータである。プログラムの性能分析、アーキテクチャの詳細な設計、ソフトウェア・ハードウェア両方の評価等を行うツールセットである。SimpleScalarLLC によって開発・保守が行われている。out-of-order 実行、投機実行、分岐予測のサポートを行った動的スケジューリングの可能なシミュレータが実装されている。また、MIPS64 互換の独自命令セットである PISA, HP 社(旧 DEC 社)の Alpha, などのプロセッサの命令セットエミュレータが既に実装されている。

4 ISIS-SimpleScalar

旧式のプロセッサシミュレータしか持たないという ISIS の欠点を改善するために、SimpleScalar が提供するシミュレータをプロセッサシミュレータとして用いる。ISIS-SimpleScalar[3] が開発され、SimpleScalar のシミュレータのうち、最近主流である out-of-order 実行や投機実行、動的分岐予測等の動的スケジューリングが可能な sim-outorder を用いる。

ISIS は C++ 言語によって実装されており、一方 SimpleScalar は C 言語によって実装されている。しかし、SimpleScalar ではグローバル変数を多用しているため、クラスに変換するのが困難である。そのため、以前、本研究室で開発された ISIS-SimpleScalar では、SimpleScalar を必要なプロセッサ数だけ namespace で覆った上で、ISIS に組み込む形で実装されている。そ

の結果，コンパイル時にプロセッサ数を決定しなければならず，バイナリサイズもプロセッサ数に応じて増加する．

5 MPI による ISIS-SimpleScalar

本研究では ISIS と SimpleScalar を別プロセスとして実行させ，SimpleScalar のプロセッサシミュレータと ISIS のプロセッサユニットを MPI を用いて接続する．このことにより，実行時にプロセッサ数を決定でき大規模なシミュレーションにも対応できると考えられる．図 1 に示す，バス結合型の共有メモリ並列計算機の構成を例に説明する．

ISIS 側 ISIS のプロセッサユニットでは，SimpleScalar と通信を行うユニットと共有メモリキャッシュへのアクセスを管理するユニットを実装している．プロセッサシミュレータとなる SimpleScalar 側のプロセスは MPI-2 で提供されている `MPI_Comm_spawn()` 関数を用いて動的に生成される．生成するプロセス数は，ISIS の実行時に引数によって任意に与えることができる．シミュレータ上で実行するオブジェクトコードは `MPI_Comm_spawn()` の引数として SimpleScalar 側に渡される．

実行中は SimpleScalar 側から要求が発生しているか監視する．要求を受信すると共有メモリキャッシュへのアクセスを実行する．読み出し要求の場合，共有メモリキャッシュからデータを取得した後，SimpleScalar 側に送信する．ISIS 側と SimpleScalar 側のクロックの進め方を管理する必要がある．

SimpleScalar 側 SimpleScalar の持つ主記憶とキャッシュは，ISIS-SimpleScalar ではそれぞれローカルメモリ，プライベートキャッシュとして利用する．シミュレータ上で実行するオブジェクトコードは，SimpleScalar のローカルメモリにロードされる．従って，共有メモリは，データに対するロード・ストアのみで使用される．共有メモリアドレスへのアクセスが発生した場合は，MPI を介して ISIS 側に送信を行う．

6 関連研究

慶應義塾大学の薬袋 俊也氏らによって，SimpleScalar を完全にクラス化した ISIS-SimpleScalar[4] の実装が行われている．SimpleScalar を ANSI 準拠の C++ により書き直し，ISIS のプロセッサユニットとして組み込んでいる．また，SimpleScalar が提供するシミュレータ `sim-outorder` をマルチプロセッサシミュレーション

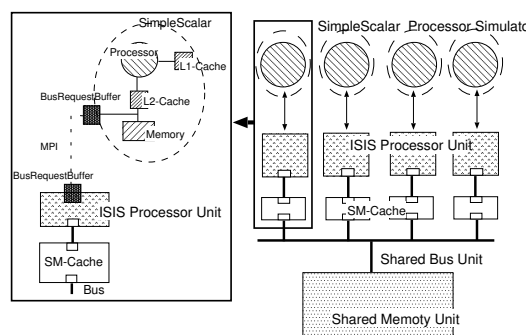


図 1: 共有メモリ型 ISIS-SimpleScalar の実装モデル

へ対応させるため，各パイプラインステージの処理の変更も行っている．

7 まとめ

並列計算機シミュレータ ISIS と，プロセッサシミュレータ SimpleScalar を MPI で接続することにより，様々なプロセッサを用いた並列計算機シミュレーションが容易に可能となる．現在実装中であり，SimpleScalar 側と，ISIS 側にある共有メモリキャッシュとの通信がボトルネックになると考えられるため，ベンチマークプログラムを用いて性能の評価を行ない高速化を検討する予定である．

参考文献

- [1] 若林 正樹，天野 英晴，並列計算機シミュレータの構築支援環境，電子情報通信学会論文誌，2001 年 4 月．
- [2] Todd Austin, Eric Larson, Dan Ernst, SimpleScalar, An Infrastructure for Computer System Modelin, Computer, February 2002 .
- [3] 高橋 瑞季，青木 聡子，埴 敏博，SimpleScalar を用いた並列計算機の実装．先進的計算基盤シンポジウム SACSIS 2004，2004 年 5 月．
- [4] 薬袋 俊也，埴 敏博，田辺 靖貴，天野 英晴，ISIS-SimpleScalar の実装，デザインガイア 2004，2004 年 12 月．