

TOPPERS 活用アイデア・アプリケーション開発コンテスト 2024

並列プログラミング言語 Elixir(エリクサー)から TOPPERS カーネル利用 C・アセンブリコードを生成するサイドチャネル攻撃防御指向コンパイラ

北九州市立大学 山崎 進

背景 1: 平易な並列プログラミング言語 Elixir(エリクサー)

提案者が研究を進めている並列プログラミング言語 Elixir(エリクサー)は、平易な並列処理と例外処理の記述を特長としており、IoT フレームワーク Nerves(ナーヴス)でも用いられている他、ウェブフレームワーク Phoenix(フェニックス)、AI フレームワーク Nx(エヌエックス)/Axon(アクソン)/Bumblebee(バンブルビー)といった強力な OSS を持っています。

たとえば Elixir でモンテカルロ法を用いて円周率を求める逐次プログラムは次のようになります(参考文献: GeekMasahiro: Elixir Flow で Ryzen の全コアぶん回したら 10 倍速かった, <https://qiita.com/GeekMasahiro/items/009fc414ca113ce391d6>, 2022).

```
1..10_000_000
```

```
|> Enum.map(fn _ -> {:rand.uniform(), :rand.uniform()} end)
```

```
|> Enum.map(fn {x, y} -> x * x + y * y < 1 end)
```

```
|> Enum.count(fn bool -> bool end)
```

```
|> Kernel./(10_000_000)
```

```
|> Kernel.*(4)
```

これを並列化したプログラムは、次のように、逐次プログラムとほぼ似たようなコードになります。

```
1..10_000_000
```

```
|> Flow.from_enumerable()
```

```
|> Flow.map(fn _ -> {:rand.uniform(), :rand.uniform()} end)
```

```
|> Flow.map(fn {x, y} -> x * x + y * y < 1 end)
```

```
|> Enum.count(fn bool -> bool end)
```

```
|> Kernel./(10_000_000)
```

```
|> Kernel.*(4)
```

なお、Apple M3 Max の MacBook Pro で実行したところ、並列版は逐次版より 12.9 倍高速に実行できました。

このような平易に並列プログラミングできる Elixir で、TOPPERS カーネルを利用するアプリケーションを記述できると、有用なのではないかと考えています。

この観点では、利用する TOPPERS カーネルの候補として、マルチプロセッサ向けである FMP3 カーネルもしくは HRMP3 カーネルが妥当そうです。

背景 2: サイドチャネル攻撃に対する体系的な防御の研究の必要性

TOPPERS カンファレンス 2024 特別講演「組込みシステムへのサイドチャネル攻撃とその対策」にもあったように、組込みシステムアプリケーションもセキュリティ脅威に晒されています。質疑において、講演講師の原先生のご専門のサイドチャネル攻撃について、コンパイラのコード生成・最適化を含めて体系的に対策を取る必要があるにもかかわらず、世界的な研究水準でもアセンブリコードレベルのサイドチャネル攻撃対策の検討にとどまっているのが現状であるとの回答がありました。そこで、こうした研究は、意義と新規性があるものと考えられます。

こうした研究を GCC や Clang など、世界的に普及した C コンパイラで行えると良いと思います。しかし実現可能性という観点に立つと、これらのコンパイラはコードベースもコミュニティも巨大であり、一朝一夕に動かすことができません。

一方で、提案者は Elixir のコード生成・最適化の研究にすでに従事しており、国内外のコミュニティの支援も受けています。サイドチャネル攻撃から防御の研究においても、Elixir のコード生成・最適化であれば、即座に始められる状況にあります。

TOPPERS カーネルにおいても、保護を重視する HRP3 カーネルと HRMP3 カーネルを利用するアプリケーションにおいて、サイドチャネル攻撃から防御するようなアプリケーションコードを記述することで、組込みソフトウェア全体のセキュリティ水準を向上させることができると考えられます。

背景 1,2 を合わせて考えると、マルチプロセッサ向けであり、かつ保護機能を備える HRMP3 カーネルを対象に研究するのが良さそうに考えています。

背景 3: Elixir を実行する方式について

Elixir を実行するための従来の方式は、母体となっている Erlang(アーラン)の処理系を移植し、BEAM(ビーム)あるいは Erlang VM(アーラン・ヴィエム)と呼ばれる仮想機械あるいは JIT によって、実行する方式が一般的です。この方式では、Erlang の長年培ってきたコードベースを再利用することができます。

しかし、背景 2 を踏まえると、コード生成・最適化を自分の手で細部まで制御する必要があります。今までの研究を踏まえると、その観点の研究に取り組むのであれば、一から処理系を構築したほうが良いと考えています。

一方、提案者は、現時点までに、Elixir のコードから直にネイティブコードを生成する技術の研究開発を進めてきています。この技術を利用して、TOPPERS カーネルを利用する C/アセンブリコードを生成するようなコンパイラを研究開発することができます。この方式ならば、背景 2 を踏まえた研究を踏まえることができます。

目的

TOPPERS カーネルプログラミングを容易にするため、平易な並列プログラミング言語である Elixir で TOPPERS カーネルを利用する C・アセンブリコードを生成するようなコンパイラを研究・開発します。

その際に、サイドチャネル攻撃から防御するようなコード生成方針を採用します。

マルチプロセッサかつセキュア指向にすることから HRMP3 カーネルを利用します。

市場性

TOPPERS プロジェクトとして、TOPPERS カーネルの利用者を拡大することが強く望まれます。Elixir の平易な並列プログラミングの記法を導入することで、その一助になることが期待できます。

また、Elixir のもつウェブや AI, IoT の OSS を TOPPERS のエコシステムに取り込め

る可能性があります。中でも、Elixir の AI 関連 OSS は Python ベースのものと互換性があり、StableDiffusion や LLM を動かすこともできます。ウェブや AI により TOPPERS の新しい市場を開拓できる可能性があります。

本提案により、サイドチャネル攻撃に耐性のあるアプリケーションを Elixir で平易に記述できるようになるのであれば、TOPPERS のセキュア RTOS としての可能性を広げることができ、新たな市場を得られる可能性があります。

研究計画

1. TOPPERS/HRMP3 カーネルのターゲットである Zybo Z7-10 を入手済なので、
Zybo Z7-10 で TOPPERS/HRMP3 カーネルを動作させる
2. TOPPERS 第 3 世代カーネル (ITRON 系) 統合仕様書に基づき、Elixir からのコード生成仕様について検討を進める
 - (ア) Elixir の send/receive にはメッセージバッファを用いて実装する
 - (イ) 固定長メモリプールとメモリオブジェクト管理機能を用いてメモリ管理を行う
 - (ウ) 時間管理機能とアラーム通知を用いて:timer モジュールと:calendar モジュールを実装する
 - (エ) 時間パーティショニング機能を用いて公平なスケジューリングを実現するよ

う努める

(オ) crypto, ssh, ssl などの暗号を用いるモジュールを実装する際には、拡張サー
ビスコール機能とメモリオブジェクト機能を用い、セキュアな実装をするよ
うに努める

3. Elixir のサブセットとなる言語仕様を定め、 2 の方針に基づき具体的なコード生成
方針を定める
4. Elixir のバイトコードを格納する BEAM ファイルからプログラムコードを読み込
み、 3 に基づいてコード変換器を実装する
5. 都度、生成されたコードの量や質に関する評価を実施する
6. 都度、得られたコードに基づいてサイドチャネル攻撃の耐性等のセキュリティ評価
を実施する
7. 徐々に言語仕様の範囲を拡大していき、既存 OSS を活用できるように整備する
8. 既存ツールチェーンを適切に運用するような、利活用しやすいツールチェーンとし
て整備していく

研究資金

2024 年度中の技術的検討に必要な Zybo Z7-10 は入手済なので、研究計画の 1～3 ま
では、追加の研究資金無しに進められます。2024 年度中に研究計画の 1～3 を地道に
進めておき、2025 年度から研究計画を立案し、研究計画の 4 以降で必要な研究資金の

獲得を目指します。

現在、研究資金の公募情報を収集しているところです。科研費については、現在の基盤 C が当面完了しないので、応募できる種目がありません。コード生成におけるサイドチャネル攻撃の研究領域として申請するのが妥当そうですが、もう少し深掘りして焦点を絞らないと、応募するのに妥当な研究資金の方向性が見えてこないと考えております。

TOPPERS プロジェクト会員から共同研究費や奨学寄付金を募るというのでも、現在の TOPPERS プロジェクト会員のニーズを踏まえないと、妥当な研究テーマが見えてこない上に、募集もうまくいかないものと思われます。本提案により TOPPERS 活用アイデア・アプリケーション開発コンテスト 2024 に応募することで、ニーズ収集を図りたいという意図があります。