

WebAssembly のメモリにおける増加量とリクエスト数の比による必要量の推定を用いたインスタンスの再起動

Restart WebAssembly instances by estimation of memory requirements per HTTP request

中川翔太¹⁾

指導教員 串田高幸¹⁾, 研究協力者 小山智之¹⁾

1)東京工科大学大学院 バイオ・情報メディア研究科 コンピュータサイエンス専攻
クラウド・分散システム研究室

キーワード: WebAssembly, 再起動, メモリ

1. 背景

WebAssembly (Wasm) は Web ブラウザ上で実行することを目的に設計された, 高速で軽量な低レベルのバイトコードである[1]. Wasm の実行単位を Wasm インスタンスと呼び, それぞれのメモリ空間は独立している.

本稿で用いる 3 つの用語を定義する. 確保量は 1 つの Wasm インスタンスが確保しているメモリの総量である. 必要量は実際に利用されているメモリ量である. 余剰量は利用されていないメモリ量である. Wasm のメモリモデルには, 必要量に応じて Wasm インスタンスの確保量は増加するが減少しないという特徴がある. 余剰量を解放するためには, Wasm インスタンスを再起動する必要があるが同時に必要量も解放する.

2. 課題

再起動により Wasm インスタンスのメモリ余剰量を削減することで, 1 つのサーバーへ収容する Wasm インスタンスの数を向上させたい. しかし, 再起動は Wasm インスタンス内部で保持されたキャッシュも同時に破棄する. そのため, Wasm インスタンスのメモリの内訳である余剰量と必要量を考慮する必要がある. また, 余剰量および必要量は直接取得できないため, 他のパラメータから見積もる必要がある.

3. 関連研究

Halvorsen らはサービスメッシュにおけるデータプレーントークンベースの認証と認可を組み込むアプローチを提案している[2]. プロキシに Envoy を利用しており, フィルター機構に Wasm を利用して認証と認可を行っている.

4. 提案方式

本提案の目的は, Wasm インスタンスの余剰量を削減しつつ, キャッシュヒット率の低下を抑制することである.

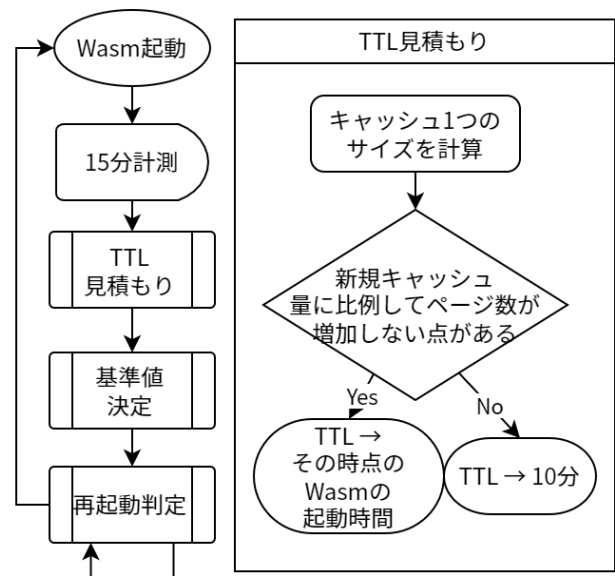


図 1 提案手法の全体像と TTL の見積もり

本提案の全体像を図 1 および図 2 に示す. 初め

に Wasm インスタンスの起動から 15 分間、再起動の判定に用いるメトリクスの収集する。

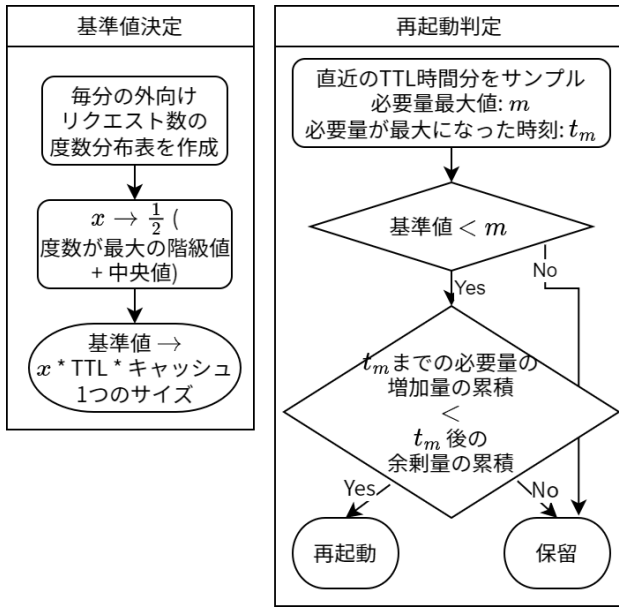


図 2 基準値の決定と再起動判定の手法

次に Wasm インスタンスの 16KB のページ単位のメモリを確保と認可サーバーへの外向けのリクエスト数の比を求める。そして、ページの確保が外向けリクエスト数よりも少なくなった場合に、古いキャッシュが破棄されたと見なし、その期間を TTL と判断する。10 分は上限値かつデフォルト値である。次に求める基準値は、再起動を繰り返しても同程度のメモリ確保量になるという見積もりの値である。再起動は、基準値以上のメモリ確保量がある場合でかつ、メモリの必要量が減少に転じ、メモリの余剰量が増加したときに実施する。これにより、有効なキャッシュの損失を抑えつつ、メモリの余剰量の削減が実現できる。

5. ユースケース・シナリオ

本稿におけるユースケースは図 3 に示すように、プロキシサーバーの管理者がプロキシサーバーへプロセスをアップストリームで動作するサービスの数だけデプロイする。サービス開発者は、自分のサービスに対応するプロキシに対して Wasm による認証・認可のためのフィルターを作成し、プロキシ上に配置する。プロキシへ配置された Wasm は HTTP リクエストに応じて、認可サーバーへ問い合わせ、その結果である認証情報をインメ

モリでキャッシュする。キャッシュには TTL があるため時間で破棄される。

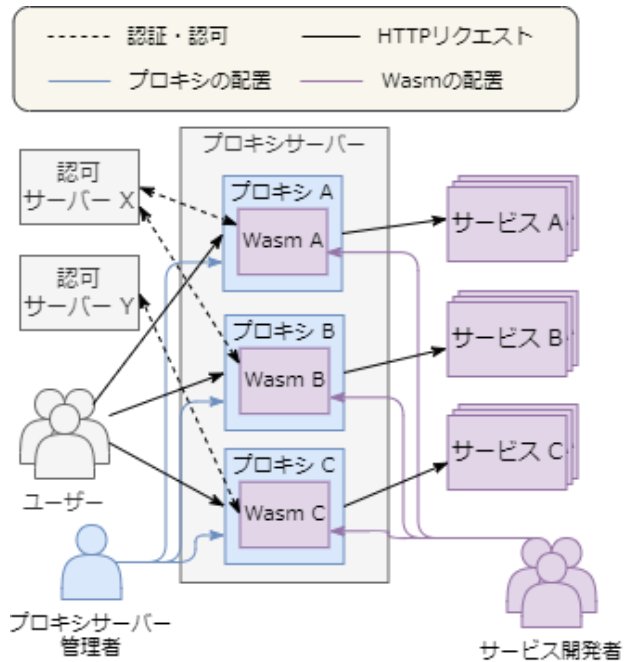


図 3 提案手法を適用するシナリオ

そのため、Wasm インスタンス上のキャッシュ量はリクエスト数にしたがって変動し、メモリの余剰量が生じる。この時、提案手法を用いることでメモリ余剰量を削減可能である。

6. 評価手法

評価手法では、Wasm インスタンス再起動を実施しなかった場合と提案手法である Wasm インスタンスの起動時間の上限値を定めそれによって再起動させた場合を比較する。同様の HTTP リクエストのシナリオを適用し、それぞれのメモリ確保量とキャッシュヒット率の推移で評価する。

参考文献

- [1] Haas, Andreas, et al. "Bringing the web up to speed with WebAssembly." Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation. 2017.
- [2] Halvorsen, Sondre. "Authentication in the mesh with WebAssembly." MS thesis. 2021.