

# AOP を用いた特化によるプロセス間通信の高速化の提案

田嶋 孝好<sup>†</sup>  
佐藤 聡<sup>†</sup>

新城 靖<sup>†</sup>  
中井 央<sup>†</sup>

## 1. はじめに

分散システムにおいてプロセス間通信は必要不可欠な要素であり、その性能の向上はとても重要な課題である。プロセス間通信に用いられる技術の1つとしてRPC (Remote Procedure Call) がある。RPC はクライアントが通常の関数呼び出しのように、サーバの手続きを呼び出すことができる技術である。RPC を用いる場合、データをクライアントとサーバ間でやり取りするには、メモリ上のデータをネットワークで転送できる形式に変換する必要がある。この処理をマーシャリング (marshaling) と呼ぶ。本研究では、RPC の種類をマーシャリングの形式により2つに分けている。1つはバイナリ形式のデータをマーシャリングするRPC、もう1つはテキスト形式のデータをマーシャリングするRPCである。本研究では前者をバイナリRPC、後者をテキストRPCと呼ぶことにする。一般にテキストRPCはバイナリRPCより低速である。

特化 (Specialization) はアプリケーションやシステムプログラムの性能を改善する手法としてよく知られている。特化を行う方法の1つとして、元のプログラムとヒントをプログラム変換器に与え、実行環境に特化されたプログラムを自動的に生成する方法がある。特化は利用者が行うため、元のプログラムの開発者は汎用的なプログラムを作成すればよい。しかし、プログラム変換器が利用できないことや、大きなプログラムでは動作しないことがある。例えば、プログラム変換器の1つである部分評価器 Tempo[1]は、C言語およびJava言語用の部分評価器であるが、大きなプログラムには対応していない。また部分評価器の開発は活発なコンパイラの開発についていくことができず、保守が行われなくなることがある。特化を手動で行うこともでき

るが、手動による特化は難しく、保守するのが困難になる。

そこで本研究では、特化したいプログラムの全体ではなく特定の部分にのみ着目し、プログラム変換器の代替手法としてアスペクト指向プログラミング (Aspect Oriented Programming, AOP) を利用することを提案する。部分評価器による特化と同様に、分散システムの利用者が実行環境に合わせてAOPを用いて特化を行う。本研究ではプロセス間通信を特化し、分散システムの性能の向上を図る。

## 2. 研究概要

本研究では、分散システムの通信部分のプログラムを特化する。具体的には、テキストRPCをバイナリRPCへ変換する。また、安全なLAN内やVPNでは、暗号化通信から暗号化を行わない通信へ変換する。本研究が対象とする分散システムおよびプロセス間通信について列挙する。

### 2.1 Microservices

Microservices[2]アーキテクチャは、独立した小さな個々のコンポーネントを組み合わせることで大きな1つの分散システムを構築する手法であり、クラウドコンピューティングにおいて近年注目を集めている。コンポーネント間の依存性が低く柔軟にソフトウェアシステムを開発することができる。各コンポーネントはREST (Representational State Transfer) でやり取りを行うことが多い。RESTは、特定のプロトコルに依存しないが、通常は通信プロトコルとしてHTTPを採用しているテキストRPCである。

AOPを用いて通信部分をテキストRPCからバイナリRPCに置き換えることで、マーシャリングが高速化され通信速度が向上すると考えられる。また個々のコンポーネントが同じノード上に存在す

<sup>†</sup> 筑波大学

る場合、通信を行わずに共有メモリを用いることでデータのやり取りを行うことができる。プロセス間通信を行うよりも共有メモリを使用する方が高速に処理を行うことができる。

## 2.2 Elasticsearch

Elasticsearch[3]は Java で記述されたオープンソースの全文検索サーバであり、REST アプリケーションの 1 つである。テキスト RPC である REST を用いてデータの登録や検索などが行える。2.1 節で述べた Microservices の特化と同様に、AOP を用いて通信部分をバイナリ RPC に置き換えることや、共有メモリを用いることで通信速度を向上できる。

## 2.3 Android

Android は Google が開発した携帯端末のためのプラットフォームである。Android 上で動くアプリケーションは Java で記述されている。

Android アプリケーションがクラウド上のサーバと通信する場合も、Elasticsearch と同様にテキスト RPC が使われていることがある。これをバイナリ RPC に変換することで CPU 処理が削減され、バッテリーの消費が抑えられると考えられる。

## 2.4 暗号化通信

分散システムで通信するデータの改ざんや盗聴を防ぐために、データの暗号化が行われる。しかし暗号化処理には時間がかかるため、暗号化が性能のボトルネックとなってしまう場合がある。

分散システムが安全な LAN 内で通信を行っている場合や、VPN を利用している場合は上位層での暗号化が不必要になる。このような環境で、AOP を用いて暗号化通信から暗号化を行わない通信に変換することで、通信速度が向上すると考えられる。

## 3. AspectJ による REST アプリケーションの特化

本研究では AspectJ[4]を用いて Java で記述された REST アプリケーションを特化する。AspectJ による特化を表したのが図 1 である。特化前のクライアントおよび REST アプリケーションに、作成したアスペクトを AspectJ 処理系で織り込み (weave) 特化したプログラムを得る。織り込みとはアスペクトをソースコードに含めてコンパイルする操作である。特化前の REST アプリケーションはクライアントと REST すなわちテキスト RPC を

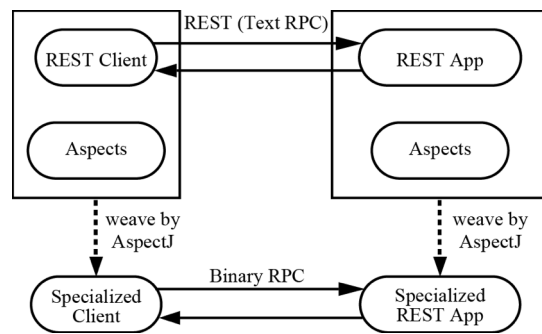


図 1 AOP を用いた REST アプリケーションの

用いて通信を行う。一方、特化後の REST アプリケーションはクライアントとバイナリ RPC を用いて通信する。

## 4. まとめと今後の予定

この論文では AOP を用いた特化によるプロセス間通信の高速化について述べた。テキスト RPC からバイナリ RPC への特化や共有メモリの利用により通信速度を高速化することで、Microservices や Elasticsearch の性能向上が期待できることを述べた。また Android について、特化により CPU 処理が削減され、携帯端末のバッテリー消費が抑えられる可能性を述べた。そして、安全な LAN 内や VPN における暗号化通信を暗号化を行わない通信に変換することで、通信速度が向上する可能性も述べた。

今後は本論文で提案した手法を実装し、性能の評価を行う。具体的には、Java で記述されたクライアントおよび REST アプリケーションを特化するアスペクトを作成する。作成したアスペクトを AspectJ 処理系で織り込み、特化前と特化後の通信速度を比較する。

## 参考文献

- [1] G. Muller, R. Marlet, E.N. Volanschi, C. Consel, C. pu, A. Goel, "Fast, Optimized Sun RPC Using Automatic Program Specialization," Proc. 19<sup>th</sup> IEEE Int'l Conf. Distributed Computing Systems (ICDCS '98), pp. 249-258, May 1998.
- [2] S. Newman, Building Microservices – Designing Fine-Grained Systems, O'Reilly Media, 2016.
- [3] Elastic, Inc., "Elasticsearch: Search & Analyze Data in Real Time," <https://www.elastic.co/products/elasticsearch/>, 2016.
- [4] The Eclipse Foundation, "The AspectJ Project," <http://www.eclipse.org/aspectj/>, 2016.