

## 発表概要

## 汎用送受信に対応したHOPEコンパイラの研究

橋本 孝太<sup>1,a)</sup> 八杉 昌宏<sup>2</sup> 平石 拓<sup>3</sup> 馬谷 誠二<sup>4</sup>

2019年7月25日発表

高生産性、スケーラビリティ、負荷分散、耐障害性は大規模な並列計算を行ううえで重要な課題である。我々が提唱中の階層的計算省略に基づく並列実行モデル HOPE はこれらの課題に取り組むものであり、複数ワーカーによる冗長実行で耐障害性を実現するとともに、階層的計算省略に基づいて台数効果を得るため、ワーカーはメッセージ媒介システムを介して部分的計算結果を交換する。HOPE 言語は HOPE ディレクティブを利用可能な拡張 C 言語である。HOPE コンパイラは、拡張 SC 言語として設計した HOPE 言語を、計算状態操作機構を追加した既存の拡張 SC 言語を経由して C 言語に変換する。その際、メッセージ媒介システムに対して部分的計算結果の読み書きを行うコードが埋め込まれる。部分的計算結果の場所は、HOPE ディレクティブの result 節で指定する。本研究では、複数の result 節の同時指定に対応したコンパイラの実装を行うとともに、汎用送受信節を言語仕様に新たに追加し、その実装を行った。汎用送受信節により、リスト構造や木構造といった複雑な構造を持つ計算結果の読み書きの指定や、読み書きにともなう処理として複数の表現を持つデータの表現を変換する処理などの指定が可能となる。たとえば、結果の読み出しでは、読み出し元となるメッセージ媒介システム内のバッファの位置や大きさを得るための一時変数とそれを用いた読み出し手順を指定する。複数 result の実装は、汎用送受信節と同様にバッファを用いて、そこに複数の部分からなる部分的計算結果を一括して含めるものとした。

## Presentation Abstract

## A Study on a HOPE Compiler That Supports Programmable Data Exchange

KOTA HASHIMOTO<sup>1,a)</sup> MASASHIRO YASUGI<sup>2</sup> TASUKU HIRAISHI<sup>3</sup> SELJI UMATANI<sup>4</sup>

Presented: July 25, 2019

High productivity, scalability, load balancing, and fault tolerance are all important issues of massively parallel computing. A “hierarchical omission”-based parallel execution model called HOPE, which we are trying to propose, addresses these issues. It realizes fault tolerance by letting workers execute the same program redundantly. In addition, HOPE workers exchange partial results via Message Mediation Systems (MMSs) to get parallel speedups by omitting hierarchical subcomputations. The HOPE language is an extended C language with HOPE directives. A HOPE compiler translates the HOPE language designed as an extended SC language into the C language via an existing extended SC language with mechanisms for legitimate execution stack access. The resulting C code reads/writes partial results from/to MMSs. Locations where partial results reside are specified using the result clauses in HOPE directives. In this study, we implemented a HOPE compiler that supports specifying multiple result clauses for a single subcomputation. In addition, we designed and implemented an extended HOPE language with programmable data exchange clauses. These clauses enable workers to read/write results with complex data structures such as lists and trees. We can also use these clauses to convert data representations prior to read/write. For example, in a clause for reading a result, we specify temporary variables for obtaining the location and size of a buffer in an MMS in which the encoded exchangeable result is stored, and a procedure to read the result using these variables. We used buffers also for implementing the multiple result clauses: multiple parts of a partial result are aggregated to a single buffer.

This is the abstract of an unrefereed presentation, and it should not preclude subsequent publication.

<sup>1</sup> 九州工業大学大学院情報工学部  
Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology, Iizuka, Fukuoka 820–8502, Japan  
<sup>2</sup> 九州工業大学大学院情報工学研究院  
Department of Computer Science and Networks, Kyushu Institute of Technology, Iizuka, Fukuoka 820–8502, Japan

<sup>3</sup> 京都大学学術情報メディアセンター  
Academic Center for Computing and Media Studies, Kyoto University, Kyoto 606–8501, Japan

<sup>4</sup> 神奈川大学理学部情報科学科  
Department of Information Sciences, Faculty of Science, Kanazawa University, Hiratsuka, Kanagawa 259–1293, Japan

a) hashimoto@pl.ai.kyutech.ac.jp