# An Evaluation of Global Address Space Languages: CoArray Fortran and Unified Parallel C

## ABSTRACT

Co-array Fortran (CAF) 和统一并行 C (UPC) 是两种用于单程序、多数据**全局地址空间编程**的新兴语言。这些语言通过为**进程间通信**而不是消息传递提供共享变量来提高程序员的工作效率。然而，这些新兴语言的性能仍有提升空间。在本文中，我们研究了 NAS MG、CG、SP 和 BT 基准变体在几种现代架构上的性能，以确定提供最佳性能必须满足的挑战。我们将这些程序的 CAF 和 UPC 变体与原始 Fortran+MPI 代码进行比较。

如今，CAF 和 UPC 程序仅在编写为使用批量通信时才能在集群上提供可扩展的性能。

然而，我们的实验揭示了 UPC 代码在所有平台上的一些重要性能瓶颈。我们解释了限制 UPC 性能的根本原因，例如同步模型、跨步数据的通信效率和源到源的翻译问题。

我们表明，它们可以通过语言扩展、新的同步结构以及后端 C 编译器的充分优化来补救。

## INTRODUCTION

人们普遍认为编写共享内存并行程序比编写消息传递程序更简单。然而，今天消息传递是在并行系统上实现性能、可扩展性和可移植性的无可争议的领导者，尤其是在商品集群上。**共享内存并行编程模型的可扩展性和性能的主要障碍是它们无法有效地利用局部性**。

出于这个原因，人们对开发用于共享内存并行编程的位置感知范式产生了相当大的兴趣。开发了用于单程序多数据 (SPMD) 并行编程的分区全局地址空间 (PGAS) 模型来解决这个问题。

在 PGAS 模型中，多个 SPMD 线程（或进程）共享其地址空间的一部分。

但是，共享空间是分区的，其中一部分对于每个线程或进程来说都是本地的。

使用 PGAS 模型的程序可以通过让每个线程或进程主要计算其本地数据来利用本地性。