# 面向非规则计算的帮助线程预取

## 摘要

非规则计算出现在许多重要的科学计算应用中，由于访存模

式的非规则性，使得非规则计算程序往往造成大量的缓存缺失，基于帮助线程的数据

预取利用空闲的硬件上下文、投机性地预取主线程需要的数据，达到延迟隐藏的目的。

本文是在共享高速缓存的多核处理器平台下，进行的面向非规则计算的帮助线程

预编译技术研究。针对现有的基于帮助线程的数据预取方法，提出了一种预编译框架，

自动化的完成线程的创建、数据的预取以及线程间的通信和同步工作。

本文首先总结了前期研究成果中的五种数据推送框架，通过分析它们的优点和局

限，提出了一种新的数据推送方法——交织同步推送方法，该方法不仅能解决不同计

算量下的数据推送问题，而且具备较低的通信和同步开销的优点。

针对访存延迟对现代处理器性能的影响， 基于片上多处理器分析与测试 了访存密集型应用程序的帮助线程数据预取性能。 结果表明热点区计算／ i# 存延迟比率对帮助线程预取性能有重大影响。 依据热点区计算／ ~5 存延迟比率合理安排帮助线程与主线程的访存任务比例时， 能达到对帮助线程性能的优化， 使帮助线程预取获得更好的性能收益。 基准测试程序的测试实验结果表明当热点区计算量很小可以忽略不计时， 帮助线程与主线程的访存任务比接近1时， 帮助线程预取获得最好的性能收 益 。

## 研究背景

为了弥补内存的延迟，人们提出了很多数据预取的方法。目前，主要的数据预取

方法主要包括软件预取、硬件预取以及软硬件结合的方法。基于预测的预取技术，利

用邻近数据被使用的概率比较大这一因数，将数据成块的预取到离处理器更近的存储

层次，达到减少访存次数，隐藏访存延迟的目的，这在一定程度上缓解了内存的压力。

尽管数据预取的方法能在一定程度上减小处理器与存储器的性能差异，但对于非规则

的数据结构，由于数据的地址不是连续的，数据预取的方法不仅预取不到合适的数据，

反而会造成 Cache 污染，不能达到降低访存延迟的目的，也不能从根本上解决“内存

墙”问题。

## 研究意义

目前，非规则计算出现在许多重要的科学计算应用程序中，由于其计算依赖关系

和访存模式的非规则性，使得非规则计算程序通常表现出极少的局部性和数据重用、

动态非连续存储访问和大量细粒度并行[34]。这类算法通过直接实现或依赖编译器的自

动优化一般很难获得高性能，特别是在并行计算机上，而基于线程的数据预取技术解

决了这个问题。

基于预取线程的数据预取技术（也称为预执行技术和预计算技术），在研究领域已

经受到了广泛的关注。相对于基于预测的预取技术，预执行技术利用空闲的硬件资源、

独立的预取线程执行长延迟的访存代码，并在主程序使用数据前将数据预取到离处理

器更近的存储层次，达到预取主线程所需数据的目的，进而减少了主线程访问内存的

次数。尤其重要的是，预取的线程先于主线程执行，并在主线程需要数据前触发缓存

缺失，从而隐藏主线程的访存延迟。

## 国内外研究现状

现在，多线程预取技术已经被公认为是一种可实现的、有效的数据预取技术。它

的实现方式是多种多样的，主要包括硬件自动预取、源代码级的软件自动预取和软硬

件结合的自动预取等方法。

基于硬件的数据预取技术已经是一项非常成熟的数据预取技术，它主要针对规则

数据执行数据预取。这里的规则数据主要是指要预取的数据相对比较集中，硬件容易

预测的数据，而对于不集中的数据（也就是非规则数据），硬件的预取效果就很不理想了，甚至会对缓存造成污染，影响程序的性能。基于硬件的预取技术的优点在于不需要程序的干预，完全由硬件实现帮助线程；缺点是需要复杂的硬件支持，难于实现。

软件自动预取技术是从源代码中抽取部分代码形成帮助线程（Source-to-Source）

并进行预执行，投机性的提前预取主线程请求的数据，减少主线程的访存次数，达到

提高程序性能的目的。目前的软件自动预取方式大多是在编译器的框架下实现的，利用编译器自动化的实现源代码的鉴别、选择、切片以及包含预取代码的推送线程的生

成过程，可以让程序员不用关心推送线程的实现方式，而致力于程序的实现。

相对于硬件预取机制来说，软件预取技术的优点在于不需要复杂的硬件支持并能

对全局的数据流和控制流进行掌控。

加利福利亚大学的 Won W. Ro 等人总结了硬件预取和软件预取的优缺点，实现了

一种混合的数据自动预取模型[18]。整个模型包含两大块：帮助线程的生成和帮助线程

的触发。他们首先利用编译器静态的实现帮助线程的生成，并对全局的数据信息和控

制信息进行掌控，提高帮助线程的预取效果；其次，利用硬件实现帮助线程的触发，

有效的降低了线程同步时的开销。Won 等人实现的软硬件结合的预取方式依赖于取指

令队列（IFQ）的大小，以及对分支预测的效率较低等缺点。

软硬件相结合的数据预取方式，不仅最大限度的降低了硬件实现的复杂度，而且

能对全局的数据流和控制流进行掌控，使得数据预取工作更加高效。

上面讲述了三种面向非规则数据的预取技术，有针对性地点出了各自的优点和缺

陷。在本文中将重点介绍基于软件的数据预取，结合前人研究工作中的优点，提出一

种面向非规则计算的多线程的预编译框架，有效地提高程序的性能。

## 帮助线程要考虑的问题

整个过程由包含特定算法的编译器自动完成。基于编译器框架的帮助线程技术需

要注意一下三个关键问题

[9]

：

1. 循环区域包含引发访存缺失（Memory Load Miss）的访存指令；

2. 帮助线程预取的数据必须在主线程需要之前预取到 Cache 中；

3. 生成的帮助线程所带来的代价应该小于帮助线程带来的收益。

整个编译器的算法主要包括访存缺失指令的识别、循环区域的选择、获取有效的

循环代码、帮助线程的生成以及线程间的通信等等。

## 实验benchmark 热点循环

## 预取粒度

数据可以以多种粒度为单位预取,单字、Cache块、内存的连续块或者程序中的数据

对象。通常情况下,预取数据的量是由底层Cache和主存系统的结构决定的。对SMP和

单处理器系统而言,Cache块是最适合的大小,而对分布式存储多处理机而言,预取更大

块的内存数据可以弥补互联网络传输数据的开销[l“]。