

并行计算模型对比分析

学号：202122280534 姓名：陈玉熙

一、引言

并行处理是实现高性能、高可用计算机系统的主要途径。并行处理技术经过多年的发展，取得了显著进步，但硬件和软件发展是不平衡的，并行软件的滞后已经严重地制约了并行机系统的应用。与顺序计算相比，并行计算的主要困难在于缺乏有效的并行计算模型。

二、并行计算模型概念

迄今为止，并行计算模型的概念仍然没有统一，存在众多的定义和分类方法。并行计算模型是并行计算机基本特征的抽象。并行计算模型不同于具体的并行计算机，并行算法的设计不能局限于某种具体的并行计算机，而必须借助抽象的计算模型。并行计算模型与并行算法设计、并行机三者之间的关系如图 1 所示。

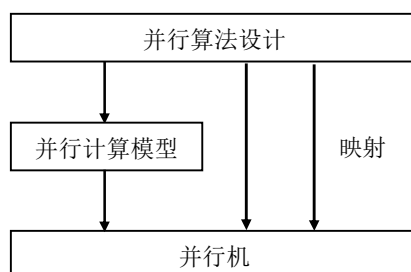


图 1. 并行算法设计、并行计算模型、并行机三者之间的关系

过去几十年来，提出了不少并行计算模型。它们的出现和发展是与并行处理系统的诞生和发展相互依存相互促进的。从硬件来看，影响并行机系统性能的因素，可概括为两方面，即并行性和局部性。我们将这两方面的分析简称为“双流”(即计算和数据流程)分析法。在并行计算机发展的初期阶段，并行机系统结构相对简单，且存储访问与 CPU 速度差异小，性能分析集中于计算，如 PRAM 模型。随着并行处理技术的发展，结构的复杂性以及存储访问速度对 CPU 计算速度差距的增大等现象，使并行计算模型不得不面对原来不被重视的性能因素。描述对象的扩大，性能因素的增多造成了当今存在的并行计算模型的多样性和复杂性。并行性是所有模型必须考虑的特征，不同的并行计算模型主要按存储访问特点来分类。如：PRAM 与共享存储类

模型 QSM, 各类网络(或抽象网络)模型(LogP)及存储层次模型(如 UMH 和 PMH)。考虑到存储与处理器计算方式则有各类修正的 PRAM 模型(如 APRAM, LPRAM 等), 异步共享存储模型(它包括一组异步处理器, 处理器之间以向共享存储空间读/写信息的方式通信), 以及各类层次模型(如 H-PRAM 等)。

二、现有并行计算模型简介

并行计算机的发展是共享存储到分布式存储、从同步到异步。与此同时, 并行计算模型的发展也经历了这个过程, 下面介绍已有的各个模型:

2.1 PRAM (Parallel Random Access Machine) 模型

PRAM 是一种理想的并行计算模型。一台 PRAM 并行计算机由若干带有局部存储器的处理机和一个全局的共享存储器构成。一个 PRAM 模型的计算由一连串的读、计算和写序列组成。在读步骤中, 每个处理器从全局内存中读取数据到局部内存中。在计算步骤中, 每个处理器处理各自局部内存中的数据, 并把结果存放在局部内存中。在写步骤中, 每个处理器将各自局部内存中的结果写入相应的全局内存中。为了解决处理机时间读写冲突, 又可按照处理机对共享单元存、取的不同约束条件进一步分为如下几种情况:

(1) EREW PRAM (Exclusive Read Exclusive Write PRAM) 模型。每次只允许一台处理机读或写某-共享单元;

(2) CREW PRAM (Concurrent Read Exclusive Write PRAM) 模型。每次允许多台处理机同时读同一个共享单元内容, 但每次只允许一台处理机向某个共享单元写内容;

(3) CRCW PRAM (Concurrent Read Concurrent Write PRAM) 模型。每次允许多台处理机同时读写同一共享单元内容。

2.2 BSP (Bulk Synchronous Parallel) 模型

根据 BSP 模型, 一个并行计算机由下面 3 部分组成: 第一, 若干个存储器或者处理机组件; 第二, 这些组件之间的点对点通信; 第三, 这些组件之间的同步机制。

为简单起见, 可以认为每个组件中包含一个处理机和本地存储器; 在模型中, 不要求关于通信系统。互连网络和同步系统的额外信息。

在 BSP 模型中, 一个并行系统由下面 3 个参数来表示:

(1) P , 系统中处理机的数目;

(2) g ，把通信开销转换为计算开销的因子；

(3) L ，两个同步之间的最短时间。

连续的两个同步之间的周期被定义为超步(super step)。在一个超步中一个处理机可以进行 3 个操作：首先各处理机处理本地存储器中的数据，可以是本地计算；然后各处理机向别的处理机提出远程内存读写请求，而通信实际发生在超步中的时间是不可预知的；最后，所有处理机进行路障同步，本次超步的数据通信仅当同步以后有效。

在 BSP 模型中，计算操作的总量用处理机在计算过程中所做的基本操作的数目来表示，通信量则用字数来表示。假设在同一次计算和通信中数据项的大小是一样的，而且假设在一个超步中的通信总量是被发送或者接受的最大字数。通过因子 g ，通信开销被转换成计算开销，其中 g 是系统中计算带宽与通信带宽的比值。具体地说，就是每个处理机发送或者接收至多 h 个字的开销与在相同的时间内进行 gh 次计算操作。 L 表示连续的两次同步之间所能执行的操作的次数，它的具体值由同步操作的实现和上层算法来决定。

同步操作使得超步之间相互无关，根据 BSP 模型实现的算法的总开销就是所有超步开销的总和，而每一个超步的开销由该超步中计算开销和通信开销所决定。

2.3 LogP 模型

LogP 是一个能很好地符合并行计算机系统中分布式存储、多处理器网络通信机制的计算模型。其由 Culler(1993)年提出的，是一种分布存储的、点到点通讯的多处理机模型，其中通讯由一组参数描述，实行隐式同步。

利用 LogP 模型，通过 4 个重要参数就可以设计良好的适应不同处理器的算法。这 4 个参数及含义如下：

(1) L (Latency) 表示源节点与接收节点进行消息传递所需要的等待或延迟时间的上限。

(2) o (overhead) 处理器发送或接收一个消息所需的处理时间开销。

(3) g (gap) 表示一台处理机连续两次发送或接收消息时的最小时间间隔，其倒数即每个处理器的有效通信带宽。

(4) p (Processor) 处理机/存储器模块个数

LogP 模型的优点在于，它将网络通信抽象为三个参数： L, o, g ，屏蔽了拓扑结构、路由算法和通信协议等细节，这点客观地反映了当代分布存储并行机系统的特性：对于实际的配置，各种拓扑结构的差别，对整个消息传输时间的影响很小，对现有并行机来说，通过轻载网络通

信时，对传输时间起决定作用的是发送和接收端处理器的开销。LogP 模型的可用性已由很多算法给予验证，并在 CM5 并行机上加以实现。

尽管 LogP 模型提出的目的之一是克服 BSP 模型的若干缺点和限制，但 LogP 与 BSP 相比，哪个更优一些。则是很难下定论的问题。有些研究者认为 LogP 模型网络容量的限制不符合现代并行机的特点，因而提出了 LogGP 模型，它比 LogP 模型多一个参数(G)以体现长消息对性能的影响。LoGPC 模型进一步将 LogGP 模型扩展，它利用 LogGP 模型描述长消息带宽，并进一步考虑了网络冲突和网络端口 DMA 对消息传递程序性能的影响。

2.4 C³ (Computation, Communication, Congestion) 模型

已有的并行计算模型，仅假设并行机内各个处理器之间以点对点的方式通信，同时这些模型也未反映网络链路或通信拥挤对性能的影响，然而对于九十年代以来广泛应用的粗粒度并行机系统：如工作站或微机机群系统，并行算法性能敏感于通信模式。S. E. Hambruch 和 A. A. Khokhar 等人提出的 C³ 并行计算模型即是一个主要用于粗粒度并行机系统的并行计算模型。此模型分析计算(computation)复杂性，通信(communication)类型和通信中出现的潜在拥挤(congestion)，并采用了一种度量来估计网络连接和拥挤对通信操作性能的影响。

C³ 模型上的计算采用与 BSP 模型类似的路障同步方式，即一个算法由若干超级步组成。每个超级步对应局部计算并跟以发送/接收信息，超级步之间进行同步。一个超级步和一个算法的性能按计算单位数和通信单位数来度量。一个超级步占用的总通信单位数由三部分组成：无拥挤时通信单位数、链路拥挤占用的通信单位数和处理器拥挤占用的通信单位数。一个超级步内无拥挤的通信单位数与选路方案、不同的发送接收原语有关。拥挤则是一个全局现象，拥挤仅依赖于处理器对之间传送的数据量，而与所用的选路策略无关。

C³ 模型用以下五个参数度量机器性能： p (处理器个数)， h (通信网络延迟)， b (通信网络的二分宽度)， s (传送一条消息的启动时间)， l (信息包长度)。

与以往的并行计算模型相比。C³ 模型更适于机群系统上粗粒度算法的开发与分析，已研究了各类并行算法，并在 Intel Touchstone Delta 机上加以验证。但 C³ 模型参数较多，对算法设计和分析有一定困难，这也是 C³ 模型未能广泛流行的原因之一。

2.5 BDM (Block Distributed Memory) 模型

目前大规模并行机系统的主流逐渐趋向于由现代高性能处理器通过互连网连接起来的分布存储并行机结构，这类结构具有很好的可扩展性，但由于其非统一地址空间的限制，编程性

较差。同时并行处理界普遍认为共享存储程序设计模型具有很好的可编程性，因而共享存储和数据并行算法应用很广泛，BDM 模型的提出者就是试图将该模型作为共享存储程序设计模型和分布存储结构模型之间的桥梁，使之既具有物理上的可扩展性，又具易编程性，同时也可用作分布存储结构上数据并行算法的性能预测。

BDM 模型所描述的分布存储并行机系统由 p 个 RAM 类处理器组成，处理器间以点对点方式进行信息传递，所有的信息传递都以包为单位，BDM 采用与 LogP 相一致的通信模式，只是增加了参数 B (信息包长度) 以体现空间局部性。BDM 假设了共享存储编程模式，对用户来讲，处理器间通信以远程存储访问请求方式完成。

BDM 用四个参数描述并行机系统： p, B, l (请求方接收到所需信息的最大延迟) 和 σ (速率：一个处理器向网络发送或从网络接收到一个字所需的时间)。BDM 模型上并行算法的复杂性由计算时间 T_{comp} 和通信时间 T_{comm} 决定。若干基于 BDM 的算法，如广播、分类、矩阵乘、FFT 等算法设计出来并在实际并行机上实现。

网络模型涉及到数据通信，目前，基于网络模型的可用程序环境逐渐统一于 MPI 和 PVM。

三、总结

上面所列出的通信模型及它们的扩展模型基本上包括了比较通用的所有的并行计算模型，对它们的设计思想和原理的了解非常有利于我们对新的模型的设计与研究。接下来对上面所说的模型做一个分析和比较。

总的来说，由于 PRAM 模型比较直观、编程简单、符合传统的程序设计概念，因而得到了广泛应用。但是在 PRAM 模型中，并行计算机被理想化了。它假定并行处理机进行的内存存取和计算都是同步操作，而且忽略了同步的开销，效率较低。而在大规模并行计算中，现有的 MIMD 机器一般是异步的，所以要在 MIMD 机器上实现 PRAM 模型是非常困难的，同步代价太高。所以，在 PRAM 上研制的并行算法不能在分布式存储系统上有效执行。

而在 BSP 模型，也是把并行计算机简化成可以进行大块同步通信的计算机。也就是说在这种情况下，算法或者程序设计人员必须把算法也设计成与 BSP 模型一致，这在某些情况下，算法不能设计成这种模式的时候就无法使用 BSP 模型。另外在 BSP 模型中，参数 g 是难以确定的，测试这些参数也有一定的难度。因为当集群实际运行的时候，网络中的通信和计算负载往往是不可预知的，这样就导致 g 的值也是不同的。同样，类 LogP 模型也有着这个问题，当一个集群中的负载不确定的情况下，这些模型都无法实际地反映系统的性能。

通过上面的分析，我们基本可以得出结论：它们都还存在着各种缺点，没有一个模型可以非常精确地刻画不同的计算模型下的网络和系统的性能。

并行计算模型将来的发展趋势很可能由下面两个方面来表现：第一，进一步可能把带宽、延迟、处理机开销等参数抽象出来，用更通用的模型来刻画集群系统的性能，就像集群式系统在带宽和延迟上的性能都比 MPP 系统更差，但是总体性能，也就是实际性能却是更好一样。这说明带宽和延迟并不能真正反映系统的性能。下一代并行计算模型可能会有更高的抽象性。第二，也许将来的并行计算模型会是与具体的计算模式相结合，针对某一个具体的应用来，某一个特殊的模型就能够非常忠实地表现出集群的性能。换句话说，就是有一种模型能够刻画出某种应用在另一种集群上的性能。这应该是下一代模型的主要发展方向。

最后，由于并行计算模型和并行式计算机的发展是相辅相成的，它们都是由实际应用所驱动的，它们的发展都要取决于新的应用的产生和发展。这也是并行计算模型的一个发展趋势。