并行计算模型*)

Models of Parallel Computation

乔香珍　刘方爱

(中国科学院计算技术研究所　北京 100080)

Abstract "To identify a small number of 'fundamental' models of parallel computation that serve as a natural basis for programming languages and that facilitate high performance hardware implementations" is one of the four Grand Challenges in parallel processing. The models of parallel computation are surveyed in this paper. The advantages and disadvantages, usability of these models are analyzed, and the research direction of this field is discussed.

Keywords Architecture, Model of parallel computation, Parallel algorithms, Performance analysis

1. 引言

为提高系统性能，并行机系统设计者在体系结构上采用了多种新技术，然而目前并行软件(包括系统软件和应用软件)的研究和开发远远落后于体系结构的进步，即体系结构上的进步并未充分反映到并行软件的设计中。相比之下，串行计算和串行软件的研究是成功的，原因之一在于有一个简单而又代表了串行计算主要性能特征的计算模型(Random Access Machines，RAM)。而并行计算缺少这样一个简单、被广泛接受并在并行计算中起同等重要作用的模型。"确定少量的并行计算模型以作为程序设计语言的自然基础，并有助于高性能硬件的实现"，这一课题仍是国际公认的并行处理中具挑战性课题之一。

由于并行计算的复杂性，往往需要在多个抽象等级上描述并行机系统。常被采用的有四类不同抽象等级的模型[1]：机器(machine)模型，体系结构(architecture)模型，计算(computational)模型和程序设计(programming)模型。机器模型属最低抽象等级，一个计算机系统按机器模型的描述包括硬件和操作系统两部分，汇编语言可视为是使用这类模型的语言。体系结构模型比机器模型抽象等级高一级，它描述存储类型、互连网及其相互作用，描述一个计算机是同步、异步，是 SIMD 类，还是 MIMD 类等等。计算模型是比体系结构模型抽象等级高一级的模型。理想来说，计算模型对一类计算机系统提供抽象描述，同时可用少量参数准确地反映该系统的资源和性能特征。这一计算模型必须简单而又充分，便于进行算法设计和分析，同时能较容易地得出问题求解所需的资源下限。不仅如此，还要求在该模型下做的算法复杂性分析能较准确地预测该类计算机系统上的性能。RAM 就是这样一个被广泛接收的串行计算模型[2]，作为冯·诺曼体系结构的计算模型，它满足了以上要求，PRAM 模型[3]是并行计算模型的一个例子。程序设计模型比计算模型的抽象等级更高一级。程序设计模型以高级抽象的观点，用形式化语言来完成对一个计算机的描述。程序设计模型和计算模型的一个关键区别在于计算模型通常按一系列存储位置来描述存储器，而程序设计模型则按数据结构描述。CSP 和消息传递模型是程序设计模型的例子。事实上，程序设计模型只是部分解决办法，为真正有用，这些模型必须实现为程序设计环境，如 MPI 和 PVM 等。

从历史上看，任何一个算法的设计和分析都开始于对欲使用的计算模型的清晰描述，使用计算模型作为算法研究基础的好处是其算法独立于某一特定技术，也便于数学描述，并行计算模型应对算法设计者、软件程序员和软、硬件厂商提供一个可信赖的开发基准。这样一个并行计算模型应提供软件和硬件之间的一个桥梁，同时象 RAM 模型在串行计算中所起重要作用一样，并行计算模型也应在并行计算中起同等重要的作用。

串行计算模型 RAM 的成功，主要得益于传统串行计算机体系结构的简单性，即冯·诺曼结构基本概括了所有传统串行机系统的结构特性。相比之下，并行计算模型的研究是困难的，由于并行体系结构的多样化，构造真正满足上述性质的并行计算模型绝非易事，多年来，国内外并行处理专家和学者一直在探索和研究。

过去二十多年来，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[4]。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[5]。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[6](LogP[7])及存储层次模型(如 UMH[8]和 PMH[9])。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[10]，LPRAM[11]等[12,13])，异步共享存储模型[14](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[1]等)。

本文主要从 PRAM 共享存储类模型、网络模型和层次(包括处理器层次和存储层次)模型三方面做介绍，它们分别对应下面的第 2、3、4 节。第 5 节给出了我们对该领域研究发展的一些看法。

*)本文得到国家自然科学基金重点项目(编号 69933020)和 863 高技术项目的资助。乔香珍　研究员。

2. PRAM 与共享存储类模型

2.1 PRAM 模型

PRAM 模型是作为并行计算的通用模型，由 Fortune 和 Wyllie[3]于 1978 年提出的。PRAM 模型可视为 RAM 模型对应的并行版本。一个 PRAM 包含 p 台通用串行处理器，全部与一个大的共享随机访问存储器 M 相连。处理器之间的通信通过共享存储器实现。在一个时间步，一个处理器可以访问（读或写）一个存储位置，或执行单个 RAM 操作。在中央主控制器控制下，所有的处理器同步地执行同一程序，因而 PRAM 的基本特性是共享与同步。在做算法分析时，PRAM 模型假设：任何一条算术运算占单位时间，但存储访问不占任何时间。PRAM 模型有多种分类法，如 EREW，CREW，ER-CW 和 CRCW 等[15]。PRAM 模型有很多优点：它适合于并行算法的表达、分析和比较，模型简单（仅用一个参数 p 即可描述），使用户能集中于并行算法设计的难点，因而文献中存在大量基于 PRAM 的有效并行算法[16,15]。PRAM 模型的主要局限在于缺少一个反映处理器间通信的机制，也没有对并行程序设计者提供存储管理和通信影响的分析，这就使得它仅仅是同步的共享存储模型。

2.2 基于 PRAM 模型的扩展模型

为克服 PRAM 的局限性，研究者提出若干基于 PRAM 的扩展模型，如 PRAM(m)[17]和 PPRAM[18]。与 PRAM 相同，PRAM(m)模型假设系统包含 p 个同步处理器，所有处理器以单位时间访问一共享存储器，但 PRAM(m)限制存储器容量为 m。共享存储器也可以视为通信介质的一个抽象，在此情况下，共享存储器的大小可解释为可同时发送和接收的通信包的数目。事实上，PRAM(m)模型假设了总体通信带宽的限制，而一般的网络模型（如 BSP[19]）假设局部通信带宽的限制（详见 3 节）。

PPRAM 模型对 PRAM 加以简单限制，它假设输入可在固定的 p 个处理器中任意划分，且共享存储器的容量为 m。因而该模型的性能可用通信带宽(m)和处理器数目(p)来分析。

早期的基于 PRAM 的扩展模型还有 DRAM(distribution random access machine)[20]，分相(phase)PRAM 模型(APRAM)[10]和局存(local-memory)PRAM(LPRAM)[11]等。

2.3 其它共享存储类模型

QSM(Queuing Shared Memory)模型是 Gibbons 提出的一个适合于描述在 MIMD 运行的体同步算法的共享存储系统模型[21]。QSM 模型包含一个共享存储器和若干处理器，每个处理器通过对共享存储位置的读/写完成通信。处理器操作由一系列同步时间段组成，而每一时间段包含三个子段：读/写和计算。该模型允许向同一个共享存储位置的读或写（但不能同时读和写）。当对一个存储位置 x 同时进行多个写时，最后一个写的内容被记入 x。一般可将 QSM 模型的共享存储器视为一组队列（queues，此即模型名称来源），每个队列对应一个共享存储位置，每一个读/写请求都需排队，任一个时刻仅完成一个请求，最大冲突值即代表了一个队列所遇到的最大延时。QSM 的计算复杂性由 p 和 g（局部操作速率和通信速率之间的比值）两个参数描述。QSM 的一个特例（g=1）是由 Gibbons 等人提出的 QRQW (Queue Read Queue Write) PRAM 模型[22]。QSM 的主要特征是共享，并加以有限的通信功能，采用大同步。若干基于 QSM 的算法已在 CRAY C90 等并行机上实现。

BSR (Broadcasting with Selective Reduction)[23]模型是 S. G. Akl 和 Stojmenovic 提出的，BSR 描述的并行机系统包含共享 M 个存储单元的 p 个处理器，在存储访问特性上与 CRCW PRAM 模型相同。BSR 的主要特征是允许广播（Broadcast）操作，这一操作包括广播、选择、归约三个步骤。在广播步，任一处理器都可发出一个广播操作；在选择步，每一存储单元按照一定的规则选取广播来的数据的一个子集；在归约步，每一存储单元对该子集的元素执行适当的归约操作，并将所得的值写入该存储单元。k>1 的 BSR 模型被称为广义 BSR 模型。可以证明，BSR 模型功能上强于 PRAM 的最强形式 combining CRCW PRAM，而又不比最弱形式的 EREW PRAM 需要更多的资源。S. G. Akl 等人在 BSR 及其广义模型上实现了若干算法。

在共享并行计算模型的基础上，研究出一系列并行程序设计环境，如 OpenMP、Posix Pthread 和数据共享语言高性能 Fortran(HPF)等等。

3. 网络模型

由于分布存储式 MIMD 并行机的发展，反映其结构特征，基于网络模型并行算法的研究在八十年代末九十年代初逐渐活跃起来，并提出了各类网络模型。

早期的网络模型分别对应具体的网络拓扑结构[15]，一般假设每个处理器带有局部存储器，但不带共享存储器。其操作可以是同步或异步的，一个异步网络是采用信息交换方式来协调其操作的。网络模型最关键的一点是将网络的拓扑结构引入模型，不同的拓扑结构，对应不同的网络模型[15]。网络结构可以真实描述相应的并行机系统，但较复杂，并过多地依赖于结构细节，不利于算法描述和分析。随并行处理技术的发展，具体的网络模型逐渐被抽象网络模型的研究所代替。

3.1 BSP(Bulk-Synchronous Parallel)模型

BSP 模型是 L. G. Valiant[19]于 1990 年提出的。它的早期版本被称为 XPRAM。它是最早地考虑到通信并以少量几个参数来描述分布存储 MIMD 系统性能的并行计算模型之一。其字面含意是“大”同步模型。Valiant 提出此模型的本意是在设计并行算法时，BSP 可作为象 RAM 模型一样的软硬件之间的桥梁，他论证了 BSP 模型可起到这样的作用，因此 BSP 也被称为桥模型。

BSP 模型所描述的并行机主要包括三个组成部分：一是 p 个计算单元，每个执行运算、存储访问操作；二是路由器，完成计算单元间点对点通信，支持这一通信的互连网的特征只以带宽参数 g 和延迟参数 L 来表征；三是同步机制，以规则的时间间隔有效地完成全部或部分计算单元间的同步。BSP 计算由一系列超级步组成。在一个超级步中，每个处理器执行局部计算，接受或发送信息。每个超级步后有一个同步，通常是屏蔽(barrier) 同步。BSP 模型所描述的并行机性能由四个参数决定：p，s(处理器速度)，L 和 g。

BSP 模型的主要优点是：如计算与通信带宽能适当地平衡，则 BSP 模型在易编程方面会呈现较大的优越性；一些重要的算法可直接用 BSP 模型实现，且 PRAM 上的并行算法可容易地移植到 BSP 模型；BSP 模型可有效地在若干互连网技术上实现。

BSP 模型的主要缺点是无论在可编程性还是性能优化方面，都希望程序具有一定的并行松驰性，且模型中的超级步长度必须能适应任意的 h-关系，这对某些算法模拟可能是一个

限制。尽管如此 BSP 仍不失为"少数几个最受欢迎的并行计算模型之一"。

文献中存在大量 BSP 扩展模型及算法库,(d,x)BSP[24]和 E-BSP[25]是其中的两个例子。(d,x)BSP 模型以两个参数扩展 BSP 模型:参数 d 表示存储体延迟,参数 x 表示存储体扩展,即存储体个数与处理器个数的比值。(d,x)BSP 模型实质是 BSP 的共享存储模型扩展。

E-BSP 模型从两方面扩展 BSP 模型:一是提供一种方式分析非均衡的通信模式;二是提出了反映网络局部性特征的一般局部性概念,以反映通信距离对性能的影响。

3.2 LogP 模型

为反映当代并行计算机主流发展方向之一的分布存储并行机所采用的关键技术,D. Culler[7]等人提出了 LogP 模型。它一提出,就受到国际并行处理界的广泛关注。

LogP 模型是一个分布存储的、点到点通信的异步并行计算模型,这一模型指明了通信网的性能特征,但不涉及互连网的具体结构。模型包括四个参数:L(从源处理器到接收点进行消息传递所需的等待或延迟);o(处理器发送或接收一个信息所需的处理时间长度);g(一个处理器连续地发送或接收一串信息时的最小时间间隔,g 的倒数对应于每个处理器的有效通信带宽);P(系统中处理器/存储器模块)。LogP 假设网络容量有限,即在任何时刻,在任何处理器上最多可接收或发送[L/g]条信息,若信息长度超过网络容量,处理器将等待,直到网络容量允许时才发送。

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[7]。

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[26]。

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[27],它比 LogP 模型多一个参数(G)以体现长消息对性能的影响。LoGPC[28]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。

3.3 C^3(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. Hambrusch 和 A. A. Khokhar 等人[29]提出的 C^3 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。

C^3 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。

C^3 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,C^3 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 C^3 模型参数较多,对算法设计和分析有一定困难,这也是 C^3 模型未能广泛流行的原因之一。

3.4 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[30]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,同时也可用来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。

4. 层次模型

我们将以并行计算为中心的层次模型和以存储为中心的层次模型分述如下。

4.1 H-PRAM 模型

H-PRAM 模型由可动态重组的同步 PRAM 组成,即一组同步的 PRAM 模型相互异步操作,一个层次关系定义了相互独立的 PRAM 的同步结构。假设有一个包含 p 个处理器的 H-PRAM,在层次的第一级,即算法的进入点,也就是计算刚开始时所看到的计算模型,是一个含 P 台处理器的同步 PRAM,并以分割指令将 p 个处理器分割为互不相连的子集—子 PRAM。此分割指令对这一模型增加了一个异步控制形式,即定义了一个二级层次关系,层次关系可以是递归分解或树等等。在 H-PRAM 模型中,每个分割步后跟以一个同步。

这一分割过程可递归进行，直到不可再分。在λ级同步的PRAM算法是由分割指令(及其相关算法)和λ+1级的子PRAM(及其相关算法)构成的，而λ=1时即得H-PRAM模型及其算法。从技术上来说，H-PRAM的子PRAM可以是除异步PRAM以外的任何一类PRAM模型，如EREW PRAM，LPRAM等等。

为分析H-PRAM上并行算法复杂性，引入了β-同步(即分割步之后的同步)的概念，文[31]讨论了H-PRAM上实现的复二叉树和FFT等算法，分析了其性能。

4.2 以存储为中心的并行计算模型

UMH(Uniform Memory Hierarchy)[8]模型是B. Alpern等人提出的，该模型提出的目的是概括计算机层次访问技术中与性能有关的特征，它是存储层次结构的一个抽象。对一个真正存储层次结构的描述(MH)是很复杂的，为实用，UMH模型作了若干简化[8]。

UMH模型将一个串行计算机的存储器视为存储容量逐渐增大的一组存储模块($M_0, M_1, \cdots$)，计算在M_0进行。这样M_0可表示一个计算机的cpu，M_1可是cache，M_2为主存储器，M_3可能是磁盘等等。总线B_u将M_u与下一个较大存储模块M_{u+1}相连，所有总线可同时工作。数据以固定大小的块为单位在总线上传输，对离计算机越远，存储容量越大的模块来说，块长度、传输一个块所需时间以及一个存储模块所包含块的个数之值越大。

UMH模型包含三个参数：α—纵横比，指一个层次存储模块M_u包含的块数n_u与其块大小s_u之比，UMH模型假设对任何一个模块M_u，α值是相同的；ρ—压缩因子，设$P_u = s_u / s_{u-1}$，UMH模型假设对任一模块M_u，该比值P_u相等，并记为ρ，即$\rho = P_u = s_u / s_{u-1}$；f(u)—传输代价函数，给出UMH中总线传输代价。按此三个参数，一个UMH模型可记为$UMH_{\alpha,\rho,f(u)}$。UMH算法的计算复杂性按传统的RAM模型分析，同时引入了通信效率的概念，以有利于存储效率分析。UMH上一个算法是否通信有效取决于f(u)。若干基于UMH模型的算法和性能分析被讨论[8]。

PMH模型[9]可视为UMH模型的并行版本。它将并行机描述为由模块组成的树，每个子女通过唯一的通道连到它的父辈。模块保存数据，叶模块也执行计算—算术运算、比较、分枝等等。一个模块的数据被分成块，块是数据在通道上传输的单位。所有的通道可以同时激活，当然不能同时移动同一数据块。PMH模型很象胖树模型[32]，但增加了显式存储模块和信息长度。在PMH模型中所有的数据移动都以父辈和子女间块传输方式进行。使用树来作并行机通信结构模型是试图在PRAM模型的简单性和任意图结构的精确性之间作一个折衷。p个处理器的PRAM模型可视为两层PMH模型的一个特例。对每个模块u，该模型有四个参数：块长度s_u，块数n_u，子女数C_u给出u所有的子女数，传输时间t_u给出在u和它的父辈之间传送一块所需的时钟周期数。一般来说，一个树的同一级上的所有模块有同样的参数。

PMH模型对一个并行机体系结构的抽象包括三步：首先选择一个树型结构来反映处理器间、处理器和存储器之间的传输带宽；然后选择块长来描述模型的通信延迟；最后用块数定义存储器容量，以及由于冲突而引起的通信性能的降低。一般分布存储系统各成份之间通信能力的分析是按延迟l和带宽b来进行的，PMH模型则以块长s和传输时间t两个参数来描述，而冲突可用限制模块中块数的方法来控制[9]。文[9]用PMH模型剖析了CM-5，KSRI[33]，IBM SP-1等并行机，并分析了性能。

类似的相关模型还有Y-PRAM[4]，D-BSP[4]，Y-BSP[4]，并行盘模型(PDM)[4]，EM-BSP*[4]，CGM[4]等等。尽管这些模型的算法研究还未广泛开展，但其思路，其研究成果对研究"被广泛接收的少量并行计算模型"这一目标有一定借鉴作用。由于目前并行体系结构日趋复杂多样，层次异构并行计算模型[34]也提了出来，这里不赘述。

小结 近二十年来，并行计算模型的研究有了巨大进展。随着商用处理器和网络技术的迅速发展，并行计算机逐渐趋向于三种统一的体系结构，即商用对称多处理机(SMP)、分布存储式并行计算机MPP和由多个SMP构成的集群系统(Cluster)。研究者从内存层次、通信网络等不同的方面，寻找改进系统性能的方法和技术。尽管新的并行计算模型不断出现[4,35]，然而真正简单而又代表当今并行机基本特点及趋势的模型还没有得到并行处理界的广泛认可，没有一个模型能够准确地刻画算法的性能在并行算法的分析方面还有较大的欠缺，因而并行计算模型的研究还是一个重要而又艰巨的任务。

综观并行计算模型发展历史与现状，有不少可借鉴之处，我们感到最重要的一条是在模型的简单性和真实反映并行机性能特征二者之间做很好的协调，同时特别要关注随技术发展仍起重要作用的因素。至今为止，没有一个并行计算模型象PRAM那样拥有大量的并行算法研究成果，近年的ICPP和ACM SPAA会议上仍有关于PRAM扩展模型和算法模拟的讨论[18,36]，这主要源于它的简单性，而并行性永远是并行计算模型中不可或缺的参数。

目前并行处理技术的发展对计算模型的研究提出了更高的要求，希望它能涵盖更多并行机的特征，如：工作站机群的通信与存储访问特征，共享分布式并行机系统[37]的特性，I/O特性，异步性及异构性等等。如何从众多、复杂的因素中忽略随技术发展而逐渐消失的次要因素，并抽取出根本性的关键要素是并行计算模型研究中的一个难点。

并行处理技术的发展和归并为并行计算模型的研究提供了有利条件。目前并行计算机发展的主流机型可粗略地归纳为两类：共享存储模式和分布存储模式，而对后者，已广泛使用可移植并行语言环境标准，如PVM和MPI。这使得并行计算模型的研究有了相对较集中的目标，同时又有算法研究可用的通用编程环境。这些条件进一步刺激了并行计算模型的发展，并为这一研究提供了有利条件。我们认为，对于SMP / NUMA(共享存储模)式并行机系统，可使用适当支持虚(virtual)PRAM[2]或修正PRAM的并行计算模型。对于分布存储类并行机系统，可在BSP，LogP和C^3等模型中做些协调，以获得一个简单而又实用的、适合于分布存储类并行机系统的并行计算模型。

参考文献

1 Heywood T, et al. A practical hierarchical model of parallel computation. J. Parallel & Distributed Computing, 1992, 16(2): 212 ~ 232

2 Krishnamurthy E V. Complexity issues in parallel and distributed computing. In: Parallel & Distributed Computing Handbook. Ed. Albert Y. Zomaya. McGraw-Hill, NewYork, 1996. 89 ~ 126

3 Fortune S, Wyllie J. Parallelism in random access machines. In:

Proc. of 10th Annual Symposium on Theory of Computing. 1978. 114 ~ 118

4 Leopold C. Parallel and Distributed Computing. John Wiley & Sons, 2001

5 乔香珍. 并行计算性能的双流分析. 计算机科学, 2001(10):7~12

6 Hambrusch S E. Models for parallel computtion. In: Proc. of the 1996 Workshop on Challenges for Parallel Processing. Aug. 1996. 92~ 95

7 Culler D,et al. LogP: Towards a realistic model of parallel computation. In: Proc. of the ACM SIGPLAN Symp. on Principles & Practices of Parallel Programming, 1993. 1 ~ 12

8 Alpern B,et al. The uniform memory hierarchy model of computation. Algorithmica, June, 1994

9 Alpern B,et al. Modeling parallel computations as memory hierarchies. In: Proc. Programming Models for Massively Parallel Computers. Sept. 1993

10 Gibbons P B. A more practical PRAM model. In: Proc. 1st ACM Symp. on Parallel Algorithms & Architectures. June, 1989. 158 ~ 168

11 Aggarwal A A, et al. Communication complexity of PRAMs. Theory of Computer Science,1990, 71(1): 3~28

12 Mansour Y,et al. Trade-offs between communication throughput and parallel time. In:Proc. 26th ACM Symp. On Theory of Computing,1994. 372~381

13 Aggarwal A A,et al. On communication latency of PRAM computations. In:Proc. 1st Symp. on Parallel Algorithms and Architectures, June, 1989. 11 ~ 21

14 Aumman Y, et al. Clock construction in fully asynchronous parallel systems and PRAM simulation. In: Proc. 33nd IEEE Symp. on Foundations of Computer Science. Oct. 1992. 147~156

15 JáJá J. An Introduction to Parallel Algorithms. Addison-Wesley Publishing Company, 1992

16 Kronsjo L I. PRAM models. In:Parallel & Distributed Computing Handbook. Ed. Albert Y. Zomaya. McGraw-Hill,NewYork,1996

17 Mansour Y, et al. Trade-offs between communication throughput and parallel time. In:Proc. 26th ACM Symp. On Theory of Computing,1994. 372~381

18 Agbaria A,et al. Communication-processor tradeoffs in limited resources PRAM. In: SPAA'99

19 Valiant L G. A bridging model for parallel computation. Communication of the ACM, 1990, 33(8):103~111

20 Leiserson C,Maggs. Communication-efficient parallel graph algorithms. Int. Par. Processing Conf. 1986

21 Gibbons P B. What good are shaed-memory models? In: Proc. of the 1996 Workshop on Challenges for Parallel Processing. Aug. 1996. 103~114

22 Gibbons P B,et al. Queue-read queue-write PRAM model: accounting for contention in parallel algorithms. In: Proc. 5th ACM-SIAM Symp. on Discrete Algorithms, Jan. 1994. 638~648

23 Akl S G,Stojmenovic I. Broadcasting with selective reduction: a powerful model of parallel computation. In:Parallel & Distributed Computing Handbook. Ed. Albert Y. Zomaya. McGraw-Hill, NewYork, 1996. 192~222

24 Blelloch G E,et al, Accounting for memory bank contention and delay in high bandwidth multiprocessors. IEEE Trans. on Par. & Dist. Systems,1997,8(9):943~958

25 Juurlink B H H,et al, The E-BSP model: incorporating general locality and unbalanced communication into the BSP model. LNCS 1124, 1996

26 Bilardi G,et al. BSP vs LogP. In: Proc. of 8th Annual ACM Symp. on Parallel Algorithms and Architectures, July, 1996

27 Alexandrov A,et al. LogGP: incorporating long messages into the LogP model for parallel computation. J. of Parallel Distributed Computation,1997,44(1):71~79

28 Moritz C A. LoGPC: modeling network contention in message-passing programs, IEEE Trans. On Par. & Dist. Systems,2001, 12(4):404~415

29 Hambrusch S E,Khokhar A A. C^3: a parallel model for coarse-grained machines. J. of Parallel and Distributed Computing, 1996, 32(1): 139~154

30 Jájá J F,Ryu K W. The block distributed memory model. IEEE Trans. on Parallel and Distributed Systems, 1996, 7(8): 830~840

31 Heywood T,et al. A practical hierarchical model of parallel computation. II. Binary tree and FFT graph algorithms. J. Parallel & Distributed Computing, 1992, 16(3): 233 ~ 249

32 Leiserson C E. Fat-trees: universal networks for hardware-efficient supercomputing. IEEE Trans. on Computing, 1985. 892~901

33 Kendall Square Research Corporation. KSR1 pronciples of operation. Oct. 1991

34 Ben-Miled,Fortes J A B. A heterogeneous hierarchical solution to cost-efficient high performance computing. Par. And Dist. Processing Symp., 1996. 138~145

35 Yan Y,Zhang X. Profit-effective parallel computing. IEEE Concurrency,1999,7(2)

36 Ulm D, Baker J W. Simulation PRAM with a MSIMD model (ASC). In: Proc. of the 1998 ICPP, Aug. 1998

37 Lenoski D E, Weber W-D. Scalable Shared-Memory Multiprocessing. Morgan Kaufmann Pub. ,1995