可扩展并行计算机系统结构和发展现状

曾庆华　陈天麒
（电子科技大学电子工程系　成都610054）

Scalable Parallel Computers: System Architecture and Up-to-Date Development

ZENG Qing-Hua　CHEN Tian-Qi
(Department of Electric Engineering, UEST, Chengdu 610054)

Abstract Scalable parallel computer is becoming a trend in developing parallel computers. Scalable computers are classified into three system models: the Symmetric Multiprocessor, the Massively Parallel Processor and the Cluster of Workstation. In this paper, the three models are discussed and Dawn parallel computers which belong to MPP and COW models are introduced.

Keywords Scalable parallel computer, Symmetric multiprocessor, Massively parallel processor, Cluster of Workstation

1 引言

自从1972年第一台并行机问世以来，并行机的发展非常快，这是由于高科技领域对计算机性能提出了越来越高的要求。随着超大规模集成电路和微处理机技术的逐渐成熟，并行机的发展也越来越快，特别是可扩展的并行机目前已经成为并行机的发展主流。

另一方面，并行计算机的飞速发展也为高科技领域提供了广阔的前景。随着计算机科学的飞速发展，现在已出现多种并行计算机系列。下面简要介绍并行机的系统结构。

大型并行机系统一般可分为六类：单指令多数据流机SIMD(Single-Instruction Multiple-Data)；并行向量处理机PVP(Parallel Vector Processor)；对称多处理机SMP；大规模并行处理机MPP；工作站机群COW和分布共享存储DSM(Distributed Shared Memory)多处理机。SIMD计算机多为专用，其余的5种均属于多指令多数据流MIMD(Multiple-Instruction Multiple-Data)计算机。目前绝大多数近代并行机均用商品硬件构成，而PVP计算机的部件很多都是定制的。

Cray C-90、Cray T-90、NEC SX-4和我国的银河1号等都是PVP并行向量处理机。这样的系统中包含了少量的高性能专门设计定制的向量处理器VP，每个至少具有1Gflops的处理能力。系统中使用了专门设计的高带宽的交叉开关网络将VP连向共享存储模块，存储器可以兆字节/秒的速度向处理器提供数据。这样的机器通常不使用高速缓存，而是使用大量的向量寄存器和指令缓冲器。

IBM R50、SGI Power Challenge、DEC Alpha服务器8400和我国的曙光1号等都是SMP对称多处理机。SMP系统使用商品微处理器(具有片上或外置高速缓存)它们经由高速总线(或交叉开关)连向共享存储器。这种机器主要应用于商务，例如数据库、在线事务处理系统和数据仓库等。重要的是系统是对称的，每个处理器可等同的访问共享存储器、I/O设备和操作系统服务。正是对称，才能开拓较高的并行度；也正是共享存储，限制系统中的处理器不能太多(一般少于64个)，同时总线和交叉开关互连一旦定型也难于扩展。

Intel Paragon、IBM SP2、Intel TFLOPS和我国的曙光1000等都是MPP大规模并行处理机。MPP一般是指超大型(Very Large-Scale)计算机系统。它具有如下特性：①处理节点采用商品微处理器；②系统中有物理上的分布式存储器；③采用高通信带宽和低延迟的互连网络(专门设计和定制的)；④能扩放至成百上千个处理器；⑤它是一种异步的MIMD机器，程序系由多个进程组成，每个都有其私有地址空间，进程间采用传递消息相互作用。MPP的主要应用是科学计算、工程模拟和信号处理等以计算为主的领域。

Stanford DASH、Cray T3D和SGI/Cray Origin2000等属于DSM分布式共享存储多处理机。高速缓存目录DIR用以支持分布高速缓存的一致性。DSM和SMP的主要差别是，DSM在物理上有分布在各节点中的局存从而形成了一个共享的存储器。对用户而言，系统硬件和软件提供了一个单地址的编程空间。DSM相对于MPP的优越性是编程较容易。

Berkeley NOW、Alpha Farm、Digital Trucluster和我国的曙光3000等都是COW工作站机群结构。在有些情况下，机群往往是低成本的变形的MPP。COW的重要界限和特征是：①COW的每个节点都是一个完整的工作站(不包括监视器、键盘、鼠标等)，这样的节点有时叫作"无头工作站"，一个节点也可以是一台PC或SMP；②各节点通过一种低成本的商品网络(如以太网、FDDI和ATM开关等)互连(有的商用机群也使用定做的网络)；③各个节点内总是有本地磁盘，而MPP节点内没有；④节点内的网络接口是松散耦合到I/O总线上的，而MPP内的网络接口是连到处理节点的存储总线上的，因而可谓是紧耦合式的；⑤一个完整的操作系统驻留在每个节点中，而MPP中通常只是个微核，COW的操作系统是工作站UNIX，加上一个附加的软件层以支持单一系统映像、并行度、通信和负载平衡等。

现在，MPP和COW之间的界线越来越模糊。例如，IBM SP2虽被看作MPP机，但它却有一个机群结构。机群相对于MPP有性能/价格比高的优势，所以在发展可扩放并行计算机方面呼声很高。SMP、MPP、DSM和COW等并行结构渐趋一致，DSM是SMP和MPP的自然结合，MPP和COW的界线逐渐不清，它们最终的结构趋向一致，形成当代并行机的公用结构。在这样的系统结构中，大量的节点可通过高速网络互

连起来。节点通常遵循着一个 Shell 结构(Shell Architecture),其中一个专门设计定制的电路将商品微处理器和其余的节点,包括板级高速缓存、局存、NIC 和磁盘连接起来。在一个节点内可有不止一个处理器。这种 Shell 结构的优点是,当处理器芯片更新换代时只需改变 Shell。

2 SGI Origin2000系列的对称多处理机

对称多处理机 SMP 结构在现今的并行服务器中几乎普遍采用,NUMA(Non-uniform Memory Access)机是 SMP 系统的自然推广,CC-NUMA(Coherent Cache NUMA)实际上是将一些 SMP 作为单节点连接起来所构成的分布共享存储系统。对称多处理机 SMP 结构主要具有以下特点:对称性、单地址空间、高速缓存及其一致性、低通信延迟。CC-NUMA 结构最显著的优点是程序员不需要明确地在节点上分配数据,系统的硬件和软件开始时自动在各节点中分配数据,在应用程序运行期间,高速缓存一致性硬件设施会自动地将数据转移至需要使用它的地方。

1996年10月,SGI 公司推出了 Origin 2000并行机系统,它采用了 CC-NUMA 结构。设计时吸取了 DASH 的经验,所设计的系统规模可以达到1024个处理器和1TB 主存,其中1-64个处理器的系统为 Origin 2000,它的结构配置属性如表1所示。

表1 Origin 2000结构配置属性

属性	立式配置	装架式配置
处理器数目	1-8	2-128
峰值速度(Gflops)	3.12	49.92
高速缓存容量(MB)	1-32	2-512
物理存储器容量(GB)	0.064-16	0.064-256
总计峰值存储带宽(GB/s)	3.12	49.92
I/O 端口数目	14	208
总计峰值 I/O 带宽(GB/s)	6.2	102

Origin 2000的每个节点包括1或者2个 MIPS R10000处理器(时钟195MHz,峰值速度395Mflops),一个可高达4MB 的高速缓存和一个 DSM 主存(物理上分散在各节点中,但所有节点都可以访问),硬件增强了基于目录的高速缓存一致性。通过一个 HUB 连接系统中的处理器、存储器、互连网络和 I/O 系统,HUB 有四个双向端口,使用195MHz 的时钟,每个口可以提供单向780MB/s 的峰值带宽和全双工1.56GB/s 的峰值带宽,它用于节点内和节点之间的通信选路,4个成对的接口电路负责内部和外部信息格式的转换。

Origin 2000系统结构设计很注重整体特性,所以存储器、I/O 和互连网络的能力都能随机器规模的扩大而成比例的增加,存储器的低延迟和通信上的高带宽使得 Origin 2000系统在市场上具有很大的优势。

3 IBM SP2系列的大规模并行机

MPP 并行机是由 SIMD 阵列机演变而来的,它沿用了 SIMD 阵列机的网络拓扑结构,但对并行机的控制方式、通信方式、通信机制、通信带宽、通信速度及操作系统等都作了根本性的改进。

所有 MPP 系统均采用物理上分布的存储器,且很多使用分布式的 I/O,每个节点有一个或者多个处理器和高速缓存、一个局部存储器,它们均连向本地互连网络,而节点之间通过高速网络相连。

现在也常把使用机群(Cluster)方法构造的 MPP 系统单列为一类,统称为 Cluster 并行机,比如 IBM SP2和下节将介绍的曙光1000A 系统、曙光2000系统等。

1994年,IBM 公司推出了 SP2并行机系统,它在结构上比较特殊,采用了 Cluster 的方法来构成 MPP。为了达到通用的目的,SP2使用了 Cluster 结构,其中每个节点都是个 RS/6000工作站,并且各有本地磁盘,每个节点内驻留一个完整的 AIX(IBM 的 UNIX 系统),各个节点经过其 I/O 总线连向专门设计的多级高速网络。SP 系列尽量使用标准的工作站组件,只有在不能满足要求时才使用专门的硬件和软件,这样的结构既简单又灵活,并且系统的规模是可扩展的。

一个 SP 系统可以包含2-512个节点,每个节点有它自己的局部存储器和本地磁盘,所有节点都连向两个网络:普通的以太网和高速交换网(HPS)。HPS 在无竞争时的硬件延迟非常小,对于512个节点仅875ns,当然实际延迟要高得多,比如一个进程发送一个空包给另一个进程至少需要40μS,这种消息传递延迟大部分是由软件开销造成的。HPS 提供的成对节点之间的双向传输带宽为40MB/s。

SP2有三种不同的节点,分别是宽节点、窄节点和高节点,它们的主要差别在于存储器的容量、数据路径宽度和 I/O 总线的不同槽数,但是所有节点都是采用时钟为66.7MHz 的 Power-2微处理器。每个处理器有一个32KB 的指令高速缓存、256 KB 的数据高速缓存、指令和转移控制单元、两个定点运算单元、两个能执行乘法-加法操作的浮点运算单元。由于定点和浮点运算可以同时进行,因此 Power-2具有267Mflops 的峰值速度。Power-2是一个超标量处理器,它使用短指令流水线、先进的转移预测技术和寄存器重命名技术,使得它在每个时钟周期内可以执行6条指令。

HPS 具有低延迟和高带宽的通信特点,它是一个多级式的网络,它提供所有节点之间的数据交换,使得无论在并行程序运行时或者主从结构数据交换时都具有最优化的性能。SP2系统提供了完整的并行化工具软件,比如 PE(Parallel Environment)、MPI(Message Passing Interface)和 PVM 等,有利于工程科研人员的研究开发。

4 曙光系列并行机

国内对并行机的研究始于80年代初,国防科技大学于1983年首先研制成功运算速度为十亿次的银河-I 巨型机,随后于92年研制成功4台运算速度为一百亿次的 MIMD 银河-II 并行机,并于93年安装在国家气象中心,用于天气预报的数据处理。

国家智能计算机研究开发中心研制的曙光1000系统是二维 MESH 体系结构的具有36个节点处理机的分布式存储 MPP 多处理机系统,其中2个为服务节点机,2个为 I/O 节点机。节点机芯片为 i860/XR,内存为32MB,节点机与二维 MESH 网的通信速率为66MB/秒,各节点机间的数据交换是基于消息传递机制并通过二维 MESH 网进行,网络通信总容量为4.8GB/秒。

4.1 曙光1000A

曙光1000系统推出后,取得了良好效果,随后又推出了曙光1000A 系统。曙光1000A 系统具有高性能、高可扩展性、高可靠性的特点,并易于使用和管理。节点机芯片为 Power-PC604,节点数可以从单个扩展到上百个。曙光1000A 吸纳了

Cluster 的思想，采用了一种比 MPP 更先进的体系结构。

Cluster 系统，是利用高速网络将一组高性能工作站或高档 PC 机，按某种结构连接起来，并在并行程序设计以及可视化人机交互继承开发环境支持下，统一调度，协调处理，实现高效并行处理的系统。

从并行角度看，Cluster 体系与 MPP 系统有很多共同点，也有一些不同。

·Cluster 体系与 MPP 系统都属于分布式存储系统，每一个节点有自己独立的存储系统，节点间通过高速网络通信。

·Cluster 体系与 MPP 系统都需要通过消息传递机制来开发并行应用。

·Cluster 体系与 MPP 系统都可以最大限度地增加节点机的个数，突破共享存储多处理机体系的瓶颈问题。

·Cluster 体系与 MPP 系统对外都表现为单一的、完整的计算机体系。

·典型的 Cluster 体系常常依赖于通用的网络体系，这样节点之间的通信效率要低于专用的 MPP 系统，因此 Cluster 体系比较适合于粗粒度、中粒度的并行。

4.2 曙光2000

曙光2000由国家智能计算机研究开发中心研制，曙光信息产业有限公司生产和销售，它基于分布式存储机群系统和消息传递体系结构，是通用的可扩展超级服务器系统。系统的节点数可以从4个扩展到128个，节点采用 PowerPC604e 微处理器，节点之间通过10/100/1000Mbps 高速以太网和高速系统网络互连。

曙光2000每个节点具有自己的微处理器、内存、内置硬盘，通过 PCI 总线与外部相连。系统具有四条互连网络，包括：定制的系统网络用于应用数据的高速通讯；内部高速以太网用于系统软件之间的数据和控制的通讯，同时是系统网络的备份；外部以太网用于连接网络用户，支持用户的登录和使用，同时是内部以太网的备份；串口网络连接到控制台上，在节点出现故障时用于诊断和恢复。系统控制台是一台独立的工作台，用来运行单一映象的系统管理软件。系统可以在选定的 I/O 节点上支持 RAID 磁盘阵列、磁带、3D 显示、Fiber Channel Disks 等 I/O 设备。

曙光2000的节点运行完整的 IBM AIX 操作系统，可以运行上千种 AIX 应用程序和数据库、网络服务等商用软件，支持 C、Fortran、Java 等程序设计语言，支持 ESSL、BLAS、Scalapack 等高效数学和工程库。系统软件提供 BCL（Basic Communication Library）底层通讯机制、定制的 PVM 和 MPI 并行程序设计环境、UE 集成化并行程序设计环境、DCDB 并行调试器、Duet 基于 Web 的联机帮助界面、Autopar 自动并行化工具、COSMOS 可扩展文件系统、NCIC-HA 对高可用性的支持、JOSS 作业管理、RMS 资源管理、CSMS 系统管理、DSC 服务器聚集软件，以及 DSM、HPF、ParaVT 等其它工具。广泛采用 JAVA 和面向对象方法，在设计上注重可用性要求。曙光2000服务器系统配置如下：

·系统共有4个节点和一个控制台，每个节点包括4个 CPU，即共有16个 CPU，可扩展到128个节点；

·CPU 主频375MHz，浮点运算峰值速度15亿次/秒；

·每个节点内存2GB；

·每个节点的高速缓存 L2 Cache：4MB；

·每个节点硬盘容量18GB；

·节点之间采用100MB 的快速以太网；

·系统支持 MPI 和 PVM 等并行计算环境。

4.3 曙光3000

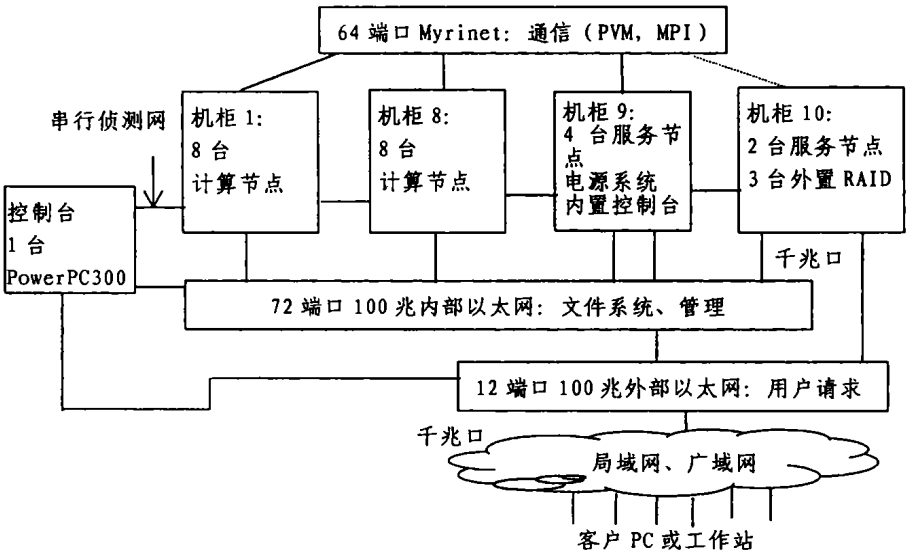

图1 曙光3000硬件体系结构

曙光3000是对曙光2000的升级。曙光3000系统由十个机柜（cabinet）、一个控制台（console），经四条网络互联而成，见图1。系统包括64个计算节点和6个服务节点（node），计算节点是4路 SMP，CPU 选用 375MHz Power3-II，服务节点是4路 SMP，CPU 选用400MHz PowerPC RS64-III，操作系统都是 IBM AIX4.3.3。系统的每个机柜中放置8个计算节点。系统中配备了三套主要网络，即系统网、内部网（100/1000Mbps 高速以太网）和外部网（10/100Mbps 标准以太网）。其中系统网采用 Myrinet 多级互连结构和交叉开关（4×4 crossbar）路由机制，该网络的 Switch Module 为2×4 结构，连接16个计算节点，具有8条互连通道，四个机柜中配备一台交换机，整个系统的最大拓扑结构是4级交叉开关互连。系统中的所有节点都与内部网相连，该网用于系统内部管理和通信，顾名思义，只有系统内部的节点可以看到这个网，其它外部节点不能直接在该网上登录；其它网络则可根据各节点的不同用途有选择地进行连接。一般，供用户登录和运行网络应用的节点要与外部

网相连，供用户运行并行程序的节点要与系统网相连，控制台只与高速以太网和标准以太网相连。整个系统通过外部网连接到 Internet 上，这样，一般用户可通过 Internet 上机，系统管理员也可通过 Internet 登录到控制台上对系统进行管理和维护(也允许系统管理员登录到系统中任何一个节点上实施管理操作)。所有节点还通过串口线连接，当内部网失效时，可以用于故障诊断。系统中节点的标准配备是9GB 的硬盘，可以在控制台上外挂各种基本的外部设备，包括光驱、磁带机和打印机，系统也根据需要在系统中的一些服务节点上外挂各种外部设备，如 RAID。用户也可增加其它外设(如三维图形显示、磁带机、打印机等)外挂到机柜9、机柜10或控制台上。

结语 本文讨论了当前最流行的三种并行计算机系统 SMP、MPP 和 COW，COW 的性能优于 MPP，所以机群技术是今后可扩展并行计算机的主流发展趋势。

参 考 文 献

1 Hwang K. Scalable Parallel Computing: Technology Architecture Programming, McGraw Hill, 1998

2 Culler D E. Parallel Computer Arichitecture. A Hardware/Software Approach. Morgan Kaufmann, 1998

3 李国杰. 曙光一号并行计算机. 计算机学报，1994，17(11)：882～889

4 Agerwala T. SP2 system architecture. IBM Systems Journal, 1995, 34(2): 154～184

(上接第154页)

只能最多包含一个运算标记，如〈EXECUTE〉、〈FUNCTION〉。

下面给出使用上述策略语言描述的一个指定经验收集过程中推荐路径长度、广度以及采纳满足某个推荐信任度等内容的本地策略，其中推荐路径的搜索深度为3，搜索广度为4，并且只有经验推荐者的推荐信任度大于0.8，其推荐经验信息才被采纳。该策略的具体描述如图3所示。

```
〈POLICY NAME="selling experience collection" TYPE="Experience Collection"〉
    〈PULE〉
        〈COLLECTION ID="selling" LENGTH="3" WIDTH="5"〉〈/COLLECTION〉
        〈FUNCTION〉
            〈GT〉
                〈FIELD ID="Recommendation Reliability"〉〈/FIELD〉
                〈CONST〉0.8〈/CONST〉
            〈/GT〉
        〈/FUNCTION〉
    〈/RUL〉
〈/POLICY〉
```

图3 本地策略描述实例

4.2 安全凭证的加密和签名

信任管理系统中的安全凭证是软件实体之间传递信任信息的载体。为了能够有效地保护安全凭证的内容在传递过程中不被篡改、伪造、泄露以及抵赖，必须在安全凭证传递之前对其进行加密和数字签名，并在凭证接收之后对其解密和签名验证。

信任管理系统中，一个安全凭证实际上表现为一个符合策略描述语言语法的 XML 文档。从经济和易实现的角度考虑，我们在安全凭证的加密和数字签名实现中遵行了安全 XML 规范[12]，并使用了一些第三方提供的符合安全 XML 规范的软件开发包。

4.3 信任管理引擎

信任管理引擎是信任管理框架的核心部分，不同于传统的信任管理系统，我们增加了信任度评估模块，其具体实现符合我们提出的基于经验的信任度计算模型。一致性验证模块的实现参考了几个已有的信任管理系统[4~6]，但我们的实现较为简单，将在以后的工作中完善。

结论 我们分析了当前软件服务协调环境下安全问题呈现的新特点和需求，指出了传统的基于策略的信任管理系统在解决上述软件环境安全问题中存在的不足，并在此基础上提出了一个结合信任度评估的信任管理框架，较好地弥补了传统信任管理系统的不足。此外，我们初步实现了一个信任管理原型系统，验证了其可行性，并体现了该信任管理框架的易操作性、合理性和策略设置的灵活性。

需要指出的是，信任管理框架的实现较为初步，有待进一步地增强策略描述语言的描述能力和一致性验证模块的证明验证能力。相信完整的系统实现，将更能体现该信任管理框架在解决软件服务协同安全问题中的优越性。

参 考 文 献

1 Herrmann P, Krumm H. Trust-adapted enforcement of security policies in distributed component-structured applications. In: Proc. of the 6th IEEE Symposium on Computers and Communications. Tunesia: IEEE Computer Society Press, 2001. 2～8

2 Khare P, Rifkin A. Trust Mangement on World Wide Web. World Wide Web Journal, 1997, 2(3): 77～112

3 Bhargava B, Zhong Y. Authorization Based on Evidence and Trust: [CERIAS Tech Report]. 2002

4 Blaze M, Feigenbaum J, Lacy J. Decentralized Trust Management. In: Proc. 17th Symposium on Security and Privacy. Oakland: IEEE, 1996. 164～173

5 Blaze M, Feigenbaum J, Ioannidis J, Keromytis A D. The KeyNote Trust Management System Version 2. Internet RFC 2704, Sep. 1999

6 Chu Y-H, Feigenbaum J, LaMacchia B, Resnick P, Strauss M. REFEREE: Trust Management for Web Applications. World Wide Web Journal, 1997, 2(2): 127～139.

7 Beth T, Borcherding M, Klein B. Valuation of Trust in Open Network. In: Proc. European Symposium On Research in Security (ESORICS). Brighton: Springer-Verlag, 1994. 3～18

8 JØsang A. A model for trust in security systems. In: Proc. of the Second Nordic Workshop on Secure Computer Systems. 1997

9 Gambetta D. Can We Trust Trust?. In: Trust: Making and Breaking Cooperative Relations. Basil Blackwell, Oxford, 1990. 213～237

10 Blaze M, Feigenbaum J, Ioannidis J, Keromytis A. The Role of Trust Management in Distributed Systems Security. In: Secure Internet Programming: Issues for mobile and distributed objects. Berlin: Springer-Verlag, 1999. 185～210

11 Povey D. Developing Electronic Trust Policies Using a Risk Management Model. In: Proc. of the 1999 CQRE Congress. Nov. 1999. 1～16

12 Ford W, Hallam-Baker P, Fox B, et al. XML Key Management Specification(XKMS). W3C Note. 2001