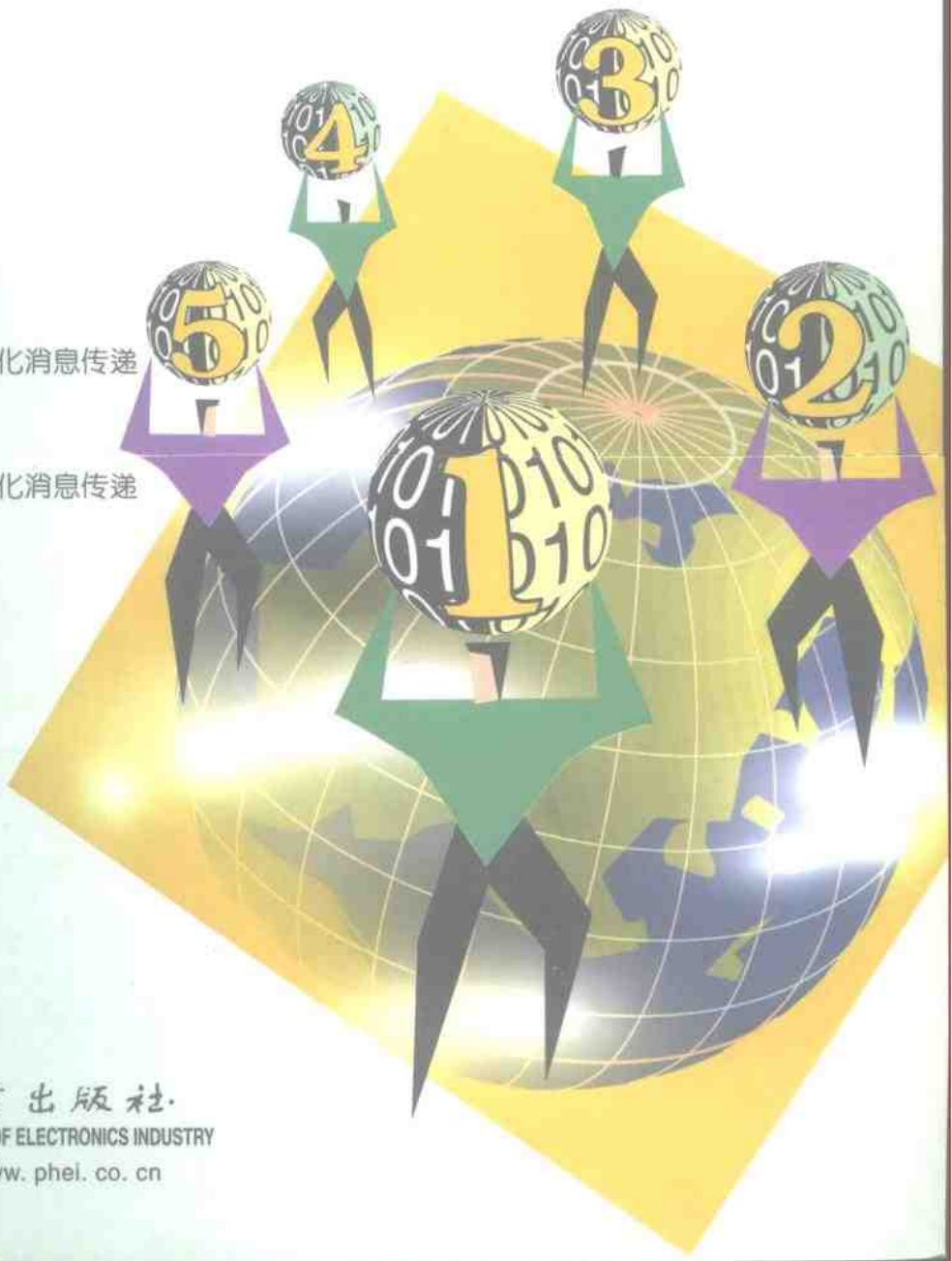
MQSeries 使用指南

# MQ Series 使用指南

IBM / Lotus 软件技术系列丛书

张进宇 编

- 队列化消息传递
- 计算框架模型中的队列化消息传递
- 在软件设计中采用队列化消息传递
- MQSeries






電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: http: // www. phei. co. cn

IBM/Lotus 软件技术系列丛书

# MQSeries 使用指南

张进宇 编


電子工業出版社
Publishing House of Electronics Industry

## 内 容 简 介

MQSeries 是 IBM 的通信中间件产品，它采用消息排队技术为应用程序提供一种跨平台的可靠的异步通信机制，是一种极具发展前景的产品。本书是 MQSeries 的通俗读物，全书共四章，第一章是基本概念，介绍消息传递和队列化技术；第二章介绍计算框架模型中的队列化消息传递；第三章介绍在软件设计中采用队列化消息传递技术；第四章介绍 MQSeries。本书内容充实，适合于 MQSeries 的初学者，使用者阅读，对开发者也有参考价值，还可作为大专院校有关专业的教材。



丛 书 名：IBM/Lotus 软件技术系列丛书
书　　名：MQSeries 使用指南
编　　者：张进宇
责任编辑：龚兰方
特约编辑：梅乃武
印 刷 者：北京科技大学印刷厂
出版发行：电子工业出版社出版、发行　URL：http://www.phei.co.cn
北京市海淀区万寿路 173 信箱　邮编 100036　发行部电话 68214070
经　　销：各地新华书店经销
开　　本：787×1092 1/16　印张：13.25　字数：339.2 千字
版　　次：1997 年 10 月第 1 版　1997 年 10 月第 1 次印刷
书　　号：ISBN 7-5053-4275-4
TP·1940
定　　价：20.00 元

# 出 版 前 言

在中国计算机技术迅速发展的今天，电子工业出版社和 IBM 中国公司软件部共同推出的 IBM/Lotus 软件技术系列丛书正式出版了。

IBM 软件产品以技术先进、性能优良、可靠易用等特点著称。最近为适应网络计算机时代的到来，IBM 软件部又推出了一系列新软件，如：MQSeries，DB2，CICS，Lotus Domino，TME-10 等，以满足广大用户的需要。这套丛书的出版，无疑是对正在翘首以望的广大读者的有益技术支持。

这套系列丛书从有利于读者理解的角度出发，介绍了 IBM 的诸多软件如：操作系统系列、Lotus Notes 系列、办公套件系列、Tivoli 网络管理系列等。每个系列产品都从入门、提高和精通三个层次展开，读者可从这种纵向、横向的网络结构中得到合适的信息。

此系列丛书的顺利出版，得到了 IBM 中国公司软件部的大力支持，众多作(译)者在成稿过程中治学严谨、认真仔细、付出了辛勤的劳动。在此一并表示衷心感谢。

尽管我们想尽力做好工作，但是由于各种因素的影响，难免有疏漏，望读者指正。

# 前　言

很高兴能有机会把 MQSeries 这样一个极具前景的产品介绍给大家。MQSeries 是 IBM 的通信中间件产品，它采用消息排队(messaging 和 queuing)技术为应用程序提供一种跨平台的可靠的异步通信机制。由于“中间件”、“消息排队”是一些较新的概念，所以本书一开始对它们作了较为详细的阐述，特别值得一提的是本书第二章介绍了一种计算模型——IBM 的开放蓝图。笔者认为它有助于为广大的计算机工作者提供一个全面的、系统的有关计算模型的概念。

本书的第三章阐述了在常用的程序设计方法中引入消息传递机制时会遇到的一些问题，针对面向功能的程序设计和面向对象的程序设计，本书给出了有关这种结合的考虑。

最后，从整体的面不只是局限于某个操作平台的角度，本书详细地介绍了 MQSeries 产品，包括其中的概念以及 MQSeries 为程序提供可靠的通信服务的机制。对于 MQSeries 提供的 MQI，本书作了完整的功能描述。

虽然本书是一本有关 MQSeries 的通俗读物，但笔者相信：在读者对 MQSeries 有一定的感性认识之后，再回过头来读这本书，同样会有更多的收获。特别是本书第四章对 MQSeries 所作的全面面详细的介绍，更有助于读者的理解。

本书的大部分内容取材于 Burnie Blakeley, Harry Harris, Rhys Lewis 合著的《Messaging & Queuing Using the MQI》。北京 IBM 公司的 Leo Liang 先生等为我推荐了这本书。同时也结合了笔者参加 IBM 在上海举办的 MQSeries 培训过程中的感受和体会。由于初次接触这种新技术，难免在本书中会有一些不足甚至错误。衷心希望本书能为 MQSeries 在中国的推广尽一份微薄的力量。借此，对 Leo Liang 先生表示感谢。

在本书成稿后，由阎小兵先生对技术与文字作了全面的校对，在此也对他表示感谢。

张进宇

# 目 录

# 第一章　基本概念

## 1.1 消息和消息传递

### 1.1.1 消息传递的发展历史

在计算机产业中，早在60年代就出现了消息传递。要对计算机科学中一些技术概念有完整的理解，就必须对该技术的发展历史有所了解。

#### 1. 早期的消息传递

队列化的消息传递对计算机产业似乎是新的概念，但自从人类商业活动一开始，人们就总能在商业信息计算处理的过程中找到这样或那样的消息传递。

① 批处理

60年代早期的穿孔卡片可看成一条消息，盛卡片的托盘可视为队列，这一盘卡片由卡片读入器读入系统，然后通过一个事先装载入系统的批处理程序进行处理。如图1.1.1所示，处理器执行批处理程序，读入数据；并把数据写入存储设备，然后通过打印机输出运行结果。

卡片是输入数据的来源。这样，批处理程序的一些工作是由一个外部信息集合（一盘卡片）的到达所引起的，而这正是基于消息驱动系统的基本特征。在那时，所有的系统都按此类方式工作。此类系统可看成是一个带有批处理器的单用户系统，数据一致性和安全性问题比较容易解决。

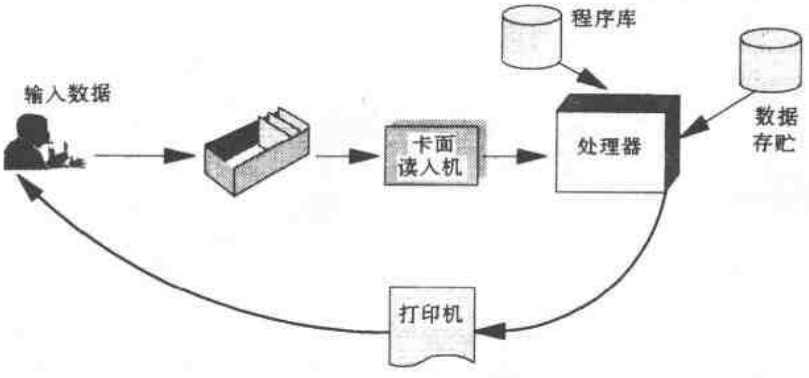



图1.1.1　批处理

在此，我们关心的是系统的输入和输出信息而不关心信息是怎样被存储在系统中的。在卡片批处理系统中，正如前面所提到的，信息被保存在穿孔的卡片上，卡片在远离系统的地方被穿孔。卡片既可以保存程序也可以保存数据。一般地，程序和数据是分开保存在不同的卡片上的，但也可以在一组卡片上既保存数据也保存程序。

输入数据在远离系统的地方被穿孔于卡片上，然后处理器在某一时间以批处理方式处理所有的输入数据。输入数据包含一个头部信息。头部技制信息定义为：哪一个程序将被执行，

执行程序所需内存在什么地方以及往什么地方、怎样输出程序执行结果。这实际上隐含了消息传递系统中一个典型的结构：一个头，其中包含控制信息，后跟处理所必需的数据。

这样的系统适合于处理大量的数据。数据一旦被处理，就能为其他用户所访问。但其缺点在于，输入数据是单独准备的，因而需要一个独立的步骤来验证输入数据的正确性。同时，用户不能快速访问被修改的数据。

② 文书工作

许多消息正是文书工作中大量信息输入的结果。

a. 面向记录的消息

批处理系统适于处理大量的数据，输入数据的准备在远离系统的地方进行。消息被组合成包含大量消息的队列，并等待预期的处理。然而，准备输入卡片的打孔机有如下一些缺点：输入卡片所包含的输入数据的有效性需额外步骤来验证。同时，大量的卡片极易被破坏或是放错顺序。这些原因最终导致了新的、更好的数据输入方式的出现。

“磁带输入”允许操作员直接向 1/2 英寸的磁带上写输入数据，然后磁带提交给系统作为输入。这样既提高了数据的输入速度又可在输入的同时进行输入数据的正确性验证并随时修改。实际上数据在最终被写到磁带上之前是存储于一小块内存中的。用磁带记录更加可靠，且占用更少的空间。但它不支持终端用户直接访问系统，也不支持多个用户对磁带上数据的共同访问。

“磁盘输入”工作站支持多重访问。数据在此类工作站中被输入到磁盘或可拆卸的硬盘上，同时，数据的可靠性也能得到验证，最后数据被传送到系统可访问的磁带上。

总之，在 60 年代有许多大的批处理系统，大量的数据操作在系统之外被执行。这类系统在许多企业中仍旧扮演着重要角色。

b. 面向屏幕的消息

60 年代，视频显示技术得到了发展，见图 1.1.2。输入的数据可放在显示设备的可修改内存中并能显示于屏幕上，然后送给系统处理。这种技术免除了传输介质（如磁带或磁盘）。根据屏幕的反馈，用户更容易纠正输入中的错误。

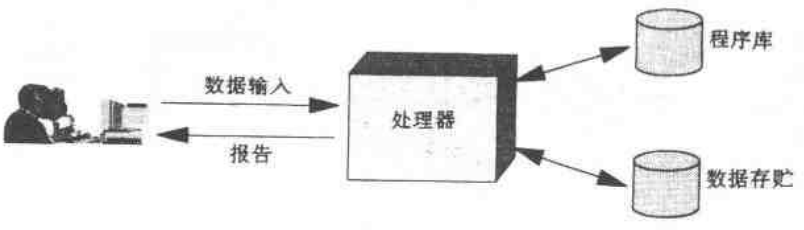



图 1.1.2 视频显示单元

视频显示设备极大地改变了系统的操作方式，用户能直接访问处理输入数据的程序，也可更加方便地访问数据。这些性能的提高也带来了一些现有系统所不能处理的问题。从此计算机技术的发展便进入了事务处理阶段。

**2. 1970～1980 年的发展**

随着通信网络，处理器能力的不断增长和直接存储设备的出现，计算机用户面对着越来越多的应用程序。许多应用程序都是基于一个“事务处理监控器”(Tramsaction monitor)，其中最值一提的是 IMS TM（信息管理系统）和 CICS TM，一个事务处理监控器控制一系列事务处理

的消息。许多年来，事务处理在企业计算中一直占主导地位。

① IMS 信息管理系统 Information Management System

今天，IMS 是一种占支配地位的数据通信和数据库系统，其根源可以追溯到美国登月计划中的一个库存跟踪系统，而且这一系统在航天工业中得到了广泛应用。该系统提供了修改在线索引数据文件的能力。这种技术在当时是很先进的。因为在此之前对文件的访问都是通过卡片穿孔、卡片读入和批处理任务来完成的。

IMS 采用的是一种事务排队处理策略。到达系统的消息包含一个事务处理代码、运行程序的名称以及运行所需要的数据。这些消息由 IMS 处理。数据被放入一个队列，每一个队列对应一个服务，在一个队列上的消息是相类似的。所引进的消息以异步的方式进行处理。这样，队列实际上提供了一个介于应用程序和 IMS 服务之间的接口。如图 1.1.3 所示。

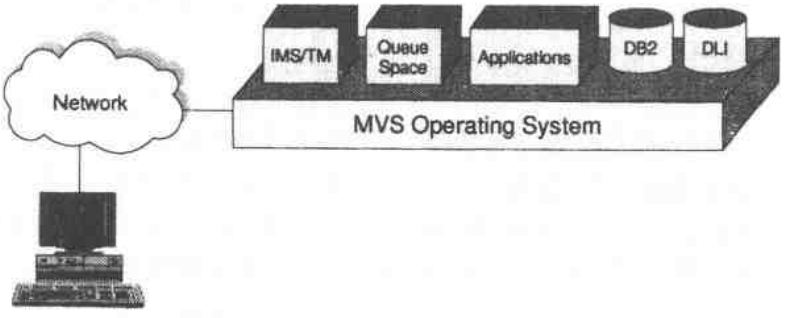



图 1.1.3 IBM 信息管理系统

事务处理是一种支持交互式应用程序的计算方式。在交互式应用中，终端用户的要求一旦被收到就会得到处理。处理的结果在一个相对较短的时间内返回给要求提出者，而事务处理系统监控着多个事务对资源的共享。事务处理系统可看作是一种向应用程序提供 ACID 特性的系统。ACID 特性定义如下：

原子性(Atomicity) 由一个事务处理引起的系统状态的改变要么都发生，要么都不发生，这些改变包括对非本系统的资源管理器管理下的资源的状态的改变。

一致性(Consistency) 一个事务处理的每一次执行都对系统状态产生同样的改变，即状态的改变是可预见和可重复的。

独立性(Isolation) 多个事务处理可同时执行，但结果(系统状态的变迁)和顺序执行一样，多个执行的效果不会产生交叉。

持久性(Durability) 一个事务被成功地执行时，所有的状态修改一旦提交，结果状态将保持直去。即使出现系统故障，该状态也能恢复。

IMS 由两个主要部分组成：

IMS/TM(Transaction Management)，大约在 1990 年以前被称为 IMS/DC(Data Communications)是系统的事务监控器，旨在执行诸如调度、授权控制、表示服务以及其他操作功能。

IMS/DB(Data Base)也就是众所周知的 DL/1，是支持层次化数据模型的数据库系统。运行于 IMS 下的应用程序可以访问 DL/1，也可以访问其他资源管理器管理的资源，比如 DB2(IBM 的关系型数据库系统)。

IMS 运行于 MVS(Multipe Virtual Storage)操作系统之上，并利用了该操作系统所提供的

服务。这意味着 IMS 利用了操作系统的进程概念，操作系统的调度，安全性服务，文件系统，程序管理等功能。IMS 同时也提供事务处理服务。这些功能被扩充和发展使得 IMS 系统能够适应大系统用户需求的增长。IMS 已经发展到能在一个事务处理系统中运行和处理多个 MVS 系统，共享庞大的、来源于成千上万个终端的数据，并具有可恢复性、安全性和不错的运行的性能。

② CICS 系统(Customer Information Control System)

在计算机产业中 CICS 是最盛行的事务处理系统。和 IMS 不一样，CICS 利用最少的由操作系统提供的服务并在单独的地址空间内实现这些功能。这意味着，在 MVS 系统中，CICS 和基于 CICS 的应用程序运行于自己的地址空间中。这种技术使得 CICS 能为自己提供的管理开销较小的操作系统功能服务，同时 CICS 应用程序环境可以较容易地移植到其他操作系统之上。所以不仅在 MVS 上有 CICS，在 ES/390 系列的 VSE(Virtual Storage Extended)和 VM(Virtual Machine)操作系统上以及 CICS/OS2，CICS/6000 和 CICS/400 上都有 CICS。现在 CICS 还可以运行于非 IBM 的平台上。

CICS 是一种事务处理环境，因而也提供 ACID 特性。但和 IMS 不一样，它并不采用排队的事务处理策略，而是直接向应用程序提供服务。应用程序在 CICS 地址空间内执行，它与 CICS 系统代码共享 CICS 地址空间，通过 EXEC CICS 接口访问系统服务，如图 1.1.4 所示的地址空间。排队的功能是由 TD(Transient Data)控制和 TS(Temporary Storage)控制提供的。TD 提供一般的队列功能。数据可临时存储以备随后的内部(CICS 系统内部)或外部进程使用。

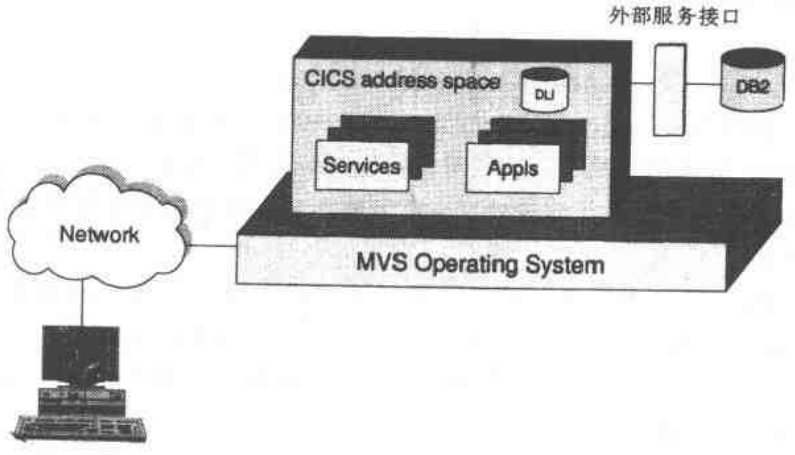



图 1.1.4 IBM 的用户信息控制系统(CICS)

在应用程序中定义的数据可以被发送到指定的目的地队列，也可以从指定的目的队列中获取，这些队列可以有以下类型：

内区(Intrapartition)队列　该队列和一个 CICS 内部功能相联系。

外区(Extrapartition)队列　该队列和一个 CICS 外部功能相联系。

内区队列具备一种称为自动事务处理激发的性能。内区队列在一个触发条件满足之后自动触发一个 CICS 事务处理。

TS 使应用程序可在临时储存队列中存储数据。这些数据可存放在内存中，也可作为辅助存放在直接存储设备上。

CICS 还在继续发展。越来越多的资源管理器的访问接口不断出现，更多的操作系统服务得到开发和利用。最初 CICS 能访问 IMS DL/1，而后已被扩展成能访问 DB2 以及安全性管理

器(Security manager)和交叉存储服务。

③ TCAM(Telecommunications Access Method)

TCAM 是 MVS 操作系统的一个子系统。该子系统允许一个远程处理(Teleprocessing)网络内的逻辑单元相互通信。为什么我们会在一个讨论队列化消息传递的章节里提到它呢？这是因为它给系统提供了一种处理消息的方法。该方法能有效地平衡输入输出性能。实际上这是一种排队机制。

一个 TCAM 系统由逻辑单元和应用程序的集合组成，并由一个 MCP(Message Control Program 消息控制程序)控制。逻辑单元位于终端上，终端与网络相连。在 MVS 操作系统控制下 TCAM 子系统被安装和运行。现在 TCAM 已不再被支持，它所实现的对逻辑单元的控制由一个新的程序，即 VTAM(Virtual Telecommunications Access Method)所取代。

在 TCAM 系统中，消息在逻辑单元间传送，而逻辑单元是系统的组成部分，如图 1.1.5 所示。TCAM 调度消息的传送，并把消息存储到目标队列中。一个目标队列和一个特定的逻辑单元或一个应用程序，或 MCP 相关联，TCAM 把相应的消息发送到该目标队列上。消息被目标队列接受的顺序由以下几个因素决定：

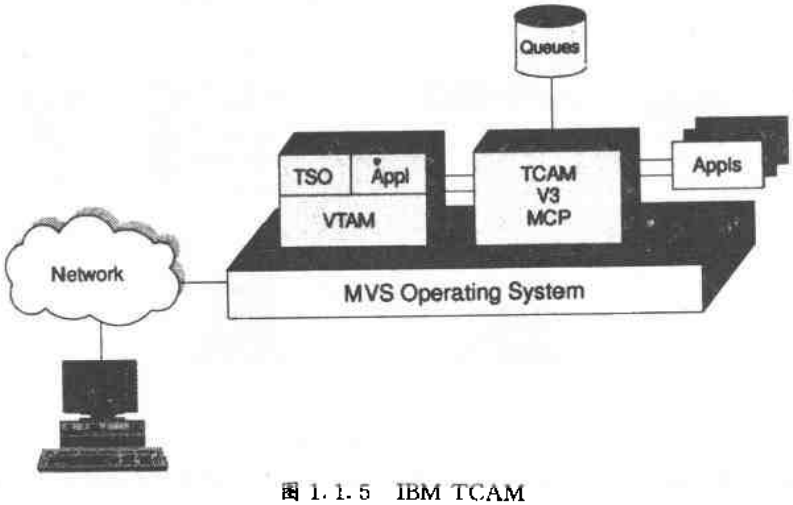



图 1.1.5 IBM TCAM

- 消息到达目标队列的顺序；
- 消息所具有的优先级；
- 在 MCP 中所使用的 HOLD 或 LOCK 宏调用。

这些队列或位于内存中或位于磁盘上，也可位于两者上。这取决于内存容量和要求的访问速度。消息由用户数据和控制数据组成，其中的控制数据由应用程序和 TCAM 创建。

正如前面所讲，消息不仅能被逻辑单元访问，也能被应用程序访问。应用程序通过 GET 或 READ 获取消息，消息被处理之后，处理结果经 PUT 或 WRITE 操作写回到队列。许多人把 TCAM 看成是当今队列化消息传递产品的前身。

④ 不断增长的网络

70 年代至 80 年代，大多数应用程序运行于单个计算机系统之上，在此类系统中，多个终端通过本地控制器和网络与中央处理器相连。网络传输一般是不可靠的。为了发现并通知应用程序网络传输出现的故障，人们制定了一些协议，而故障的排除和通信的恢复一般是应用程序的工作，这大大加重了应用程序的负担。同时也导致了高度集中控制的计算机层次体系结构。这种结构不能处理网络上出现的大量错误。正是在这样的环境中，IBM 的 SNA(Systems

Network Architecture)得到了发展。它引起了以工作站为基础的网络的增长,后来程序之间也可以基于 SNA 进行通信。

大多数通信发生在一个企业内部,但随着应用程序覆盖范围的增长,分布式应用的出现和访问设备性能的提高,迫切需要提高网络访问的速度和网络的可靠性。WAN(Wide Area Networks)和 LAN(Local Area Networks)变得更加不同。于是出现了不同的协议以满足 WAN 或 LAN 中不同的应用环境。

⑤ 局域数据中心

70 年代早期,系统的性能价格比不断提高。内存不断增大,系统体系结构逐步改善。随着事务处理系统的出现,系统越来越需要完整、精确的数据,这导致了许多分布式数据中心的建立。如图 1.1.6 数据在各个地方被收集和处理。然后,在一定的时间内大量的包含有数据的磁带被运输到计算中心。

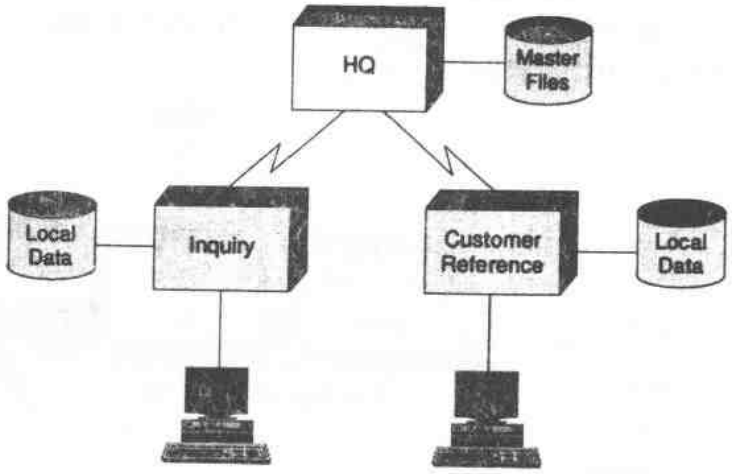



图 1.1.6 数据中心

随着计算中心的不断发展,一些专门的系统被开发并用来处理数据输入,查询和结果输出。这些系统明显地提高了系统的整体性能。但因为它们比较专门化,对系统功能的修改往往涉及到修改硬件设备和操作系统。

⑥ 个人计算机

随着越来越多的公众对计算机的使用,出现了个人计算机。

在这之前,计算机系统都是既庞大又昂贵,一般为大型企业或团体所有。但对中等规模的计算能力的使用和控制是可能的。人们能收集并共享他们的数据,有着各自的计算需求。许多公司甚至用户本身开始基于便宜的处理能力开发一些小的、能完成一定功能的应用程序,这意味着以往的数据中心的功能被延伸而分布在大量的的个人计算机中。

⑦ 客户-服务器计算

当许多个人机在运行各自的应用程序的同时,它们的另一项功能就是与大型机以及大型机所附带的数据中心相连,从而向应用程序提供一个功能更强的开发和运行环境。

现在已进入客户-服务器计算阶段。在这个阶段,工作站向用户提供一个友好界面并处理用户的要求,而大型计算机提供对数据的存储和访问。这儿所指的大型计算机既可以是一个通过 LAN 相连的文件服务器,也可以是一个通过 WAN 相连的某一企业的处理机。客户-服务器

模型的实质是一个小的计算机向一个大的计算机申请服务。

然而，在某些情况下，客户和服务器并非总能被清楚地定义。比如：一个在主机上运行的应用程序试图访问一个通过局域网(LAN)相连的打印机，该打印机隶属于一台个人机。此时，主机上的应用程序是客户，而这台个人机就成了服务器。同时必需在商业事务需要的地方提供需要的服务，这一目的使得为不同的应用程序提供对等通信成为必要。

最初，一个应用程序运行于一个系统之上，而今天，应用程序之间需要相互通信，这种程序间的通信，在大型机内部早已实现。但现在的通信发生在不同的工作站之间。客户-服务器的解决方案需要的是分布式的管理和操作。关于分布式环境将在后面的章节介绍。正是由于应用程序之间需要相互通信才使队列化的消息传递越来越受到人们的重视。

⑧ 程序间的消息传递

70 年代和 80 年代的事务处理监控器允许程序相互直接通信。但通信的程序必须由同一个监控器控制。通信协议和被传递消息的格式由监控器定义。当运行于不同的系统之上的程序需进行通信时，以上所提到的事务处理监控器就不能胜任了。这种通信需要新的技术。

到了 80 年代，这种无需由人干预的程序间通信变得越来越必要。它需要在应用环境中具备新的控制方法和恢复手段。因为应用程序不再能依靠操作员来处理通信故障。

⑨ 队列化消息传递的应用程序

当事务处理成为许多企业应用的基石的时候，人们越来越认识到队列技术的价值。一些基于排队技术原则的子系统被开发而成。这些子系统(常不为人所知)往往成了系统的基本构成。这意味着许多企业正在使用了本书介绍的技术，也许读者已经熟悉某个使用了本书介绍的技术的消息传递系统。

分布式应用环境 DAE(Distributed Application Environment)可看成是一个基于队列的消息传递应用系统。DAE 提供一个支持各种功能的应用环境，这些功能使得人们能在不同的制造商的设备上开发应用程序并且程序间能通信。DAE 环境由分布式网络环境中的资源组成。该环境既可被看成一个节点，也可被看成一个分布式网络。应用程序通过一个 API 接口申请服务。服务位于应用程序之外，应用程序通过一个消息传递接口与服务提供者松散地相连。服务包括：

- 基本系统服务：这些服务控制分布式应用环境，并提供独立于操作系统的系统服务。
- 通信服务：允许应用程序通过无连接模式相互通信。
- 数据服务：在多用户环境下提供高效的数据组织和数据检索。
- 用户服务：提供一致的表示服务。
- 设备服务：提供物理设备的逻辑抽象。

基本系统服务中包含一个消息排队传送机制，该机制提供一种基于优先级的磁盘排队功能。对磁盘的本地或远程访问对应用程序是透明的。

### 3. 90 年代

90 年代充满了挑战，如图 1.1.7，为数繁多的网络协议，众多的设备制造商以及各种不同类型的应用程序使得程序间通信更加困难。

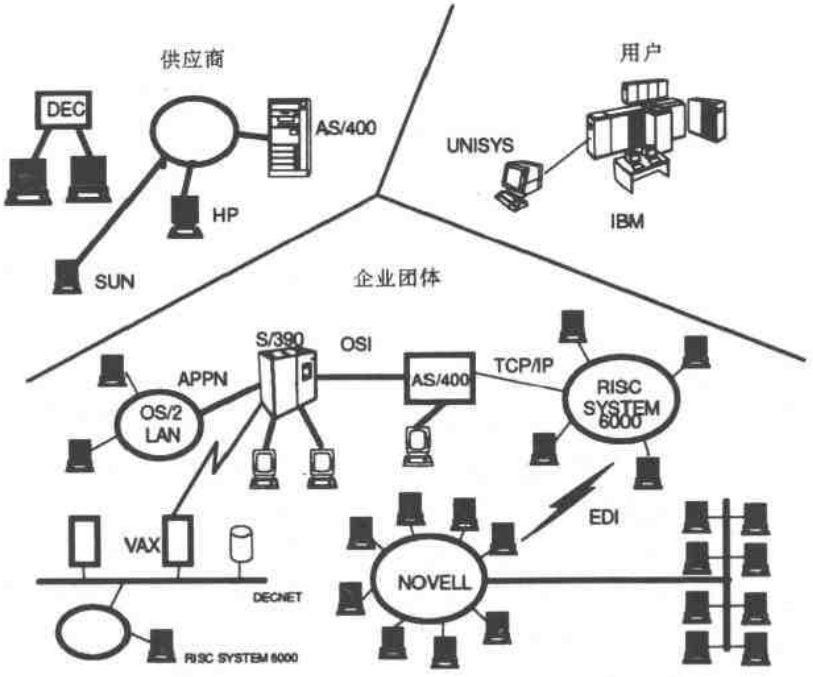



图 1.1.7 90 年代的挑战

① 队列化消息传递

使用排队技术的系统很早就出现了,IMS 就是其中之一。我们也看到跨越不同操作环境的通信变得越来越必要,一些系统虽然已把队列化消息传递纳入其基础结构中,但这些功能却不能被应用程序开发人中所访问。

② 开放的网络互连

在今天的企业中,经常包含多个数据中心,各个部门的应用程序以及来源于不同制造商的硬件设备,甚至不同的操作系统。这些不同的应用程序和各个数据中心的信息对企业的高效运作特别重要,而我们怎样来访问它们呢?

一种解决方案就是逐渐过渡,最后只使用一种操作平台,而把其他的应用和数据移植于其上。但这开销太大而且需要一个较长的过程。另外一种策略就是允许现有的应用程序能够相互通信,交流信息,而不管应用程序运行于什么样的操作平台之上。这种策略无需在企业内部只运行一种操作平台和使用一种通信协议。它只要求应用程序间的通信机制能工作于众多的操作平台和通信协议之上。

### 1.1.2 消息传递的目的

同今天计算机产业界流行的同步通信技术相比,消息传递是异步的,同步递信要求通信双方和递信资源随时可用。

#### 1. 引言

历史上有种类繁多的消息传递的方式,但消息传递的目的却只有一个:从一个地点到另一个地点的可靠的信息传递。这既是人力传送也是高技术卫星通信的目的。本章我们讨论机器

辅助的通信和消息传递，比如使用电话、传真和计算机等设备的通信。

我们已经看到消息很早就出现于计算机产业中，但消息的使用方式和消息传递的方式却发生了很大的变化，那些包含大量但简单的信息处理的商业应用，比如航空订票系统和银行查询系统促进了消息传递的发展。在这类系统中，消息传递的基本目的是向用户提供快速的信息检索，数据传送带有可预料的可靠性并且需人工干预进行故障恢复。而今天，各种复杂的商业信息必须在没有人工干预的情况下被经常地交换。不使用消息传递机制的基于连接的解决方案引起了“连接爆炸”。导致了很长的系统重启时间。这是当今商业环境所不能接受的。因此无连接的消息传递得到了逐步发展。

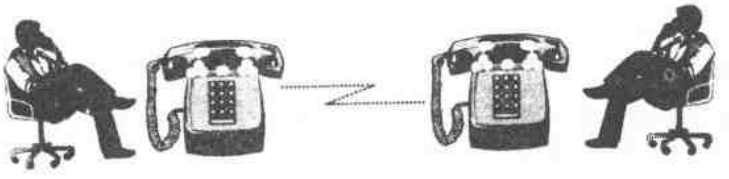
我们先来看一看所说的异步和无连接消息传递以及一些通信的实例。

2. 人与人的消息传递

消息在程序间传递之前很早就在人与人之间传递。人与人之间的消息传递的速度是“人类速度”(秒，分钟，小时，天)；程序之间的消息传递速度是通信网络的速度(毫秒，秒)。

① 电话

电话(图 1.1.8)给人类通信方式带来了极大变化，人们不再需要物理接触，通信也不受地域和国界的限制。由于我们不能看见通话的对方，一次成功的通话需要：建立一种协议；通话双方同时在场；电话线连接随时可用。这种协议的内容包括通话双方说同样的语言吗？谁先说话？在提出问题和发表评论之前应该等多久才不致于打断对方的反应？在相互通信的计算机之间也需建立类似的协议。



成功的电话通信依赖于：

通信线路随时可用

打电话的双方必须在场

使用共同的语言

图 1.1.8　电话

借助于电话和现有的通话协议建立了全球范围内的电话网络。问题是：如果通话的对方不在怎么办？

②电话应答机

这种技术回答了上面的问题。接收者能异步地处理电话消息。应答机使通话双方即使不都在场也能有效通信。这实际上引入了时间独立性特点。消息能在需要和可能的时候被处理。这实际上是人与人之间的队列化消息传递。

电话和电话应答机都是语音通信的手段，具备相应的传送机制的文字也可作为异步通信的有效手段。比如，信件、书和邮政早于电话之前就已建立，不同的是信息传送的速度。电话传送信息的速度大大快于信件和书。但技术的发展把这种速度赋予了文字的传送。这就导致了传真机的出现。

③传真