

# 消息队列中间件系统中消息队列与消息分发技术研究

陆 庆, 周世杰, 秦志光, 刘锦德

(电子科技大学 计算机科学与工程学院, 四川 成都 610054)

**摘 要:** 从中间件的基本消息流入手, 简单介绍了消息队列中间件中消息分发机制以及消息分发机制消息流过程。结合消息分发机制, 讨论了队列式共享内存管理在中间件系统中的重要性。提出了消息队列中间件系统中消息分发器与消息队列的各种组合关系, 描述了各种形式下的消息流过程, 对每种方式作了详细分析和讨论, 总结了各自的特点。最后, 比较了它们对消息队列中间件效率、服务类型、消息优先级等的影响。

**关键词:** 中间件; 消息队列中间件; 共享内存

中图法分类号: TP393

文献标识码: A

文章编号: 1001-3695(2003)08-0051-03

## Study on Message Queuing and Message Dispatching in the Message Middleware

LU Qing, ZHOU Shi-jie, QIN Zhi-guang, LIU Jin-de

(College of Computer Science & Engineering, University of Electronic Science & Technology, Chengdu Sichuan 610054, China)

**Abstract:** The data processes in middleware are analyzed. Then, two allocation policies, static allocation and dynamic allocation, in shared memory are referred. After discussing the importance of management of shared memory in middleware system, all kinds of relationship between message dispatcher and message queue adopted by shared memory management technology are put forward. Finally, the characteristics of all kinds of message dispatcher and message queue technology are summarized and compared.

**Key words:** Middleware; Message Queue Middleware; Shared Memory

### 1 引言

中间件系统是在客户/服务器模式下用于应用程序之间可靠、高效、安全地交换信息的软件实体。其基本结构如图 1 所示<sup>[1]</sup>。

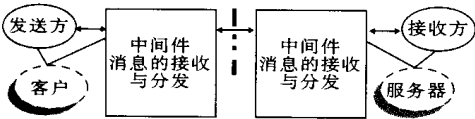


图 1 中间件的信息交换逻辑

由图 1 可见, 当发送方应用程序将消息发送到中间件客户端后, 中间件消息接收进程(可以称之为消息接收器 MR, Message Receiver)执行消息接收操作(Message Receiving)。如果消息接收成功, 与之合作的消息分发进程(可以称之为消息分发器 MD, Message Dispatcher)通过消息通道(如果消息通道是网络, 则一般采用 Socket 方式)将消息分发(Message Dispatching)到与之对应的中间件服务器端。服务器端的消息接收器收到合作方传来的消息后, 必须以某种方式启动与之合作的消息分发器, 从而将这些消息按一定的方式传送到对应的接收方

应用程序, 进行消息处理(比如写数据库)。当消息处理结束后, 中间件还要负责将处理结果的信息返回发送方应用程序。对于中间件消息接收器与消息分发器之间的信息交换, 有许多技术可以提供良好的解决方案, 而对于消息分发器与应用程序之间的信息交换(消息分发), 则实质是向应用程序提供统一编程接口的问题, 因而难度较大。

### 2 消息队列中间件中消息的分发机制

从工作定义的角度来讲, 消息队列中间件是利用消息队列技术作为其通信手段的中间件系统。它的基本消息流与一般中间件系统相似。图 2 是消息队列中间件服务器端的消息流示意图。

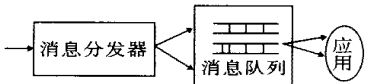


图 2 消息分发器与消息队列

我们之所以讨论服务器端, 是因为在多对一(多个客户程序, 一个服务程序)的模式下, 服务器端的消息分发器与消息处理应用程序之间消息交换的效率是整个系统的瓶颈<sup>[1-8]</sup>; 至于客户端, 则与此相似。由图 2 可

见,在消息队列中间件中,服务器端消息分发器与应用程序之间的消息交换是依靠消息队列进行的<sup>[1-3,8]</sup>,其中消息分发器、消息队列以及它们之间如何相互协调使消息传递流畅,在消息队列中间件中占有举足轻重的地位。

消息队列中间件中的消息队列,以共享内存式队列的效率最高,并且因其也有使用灵活、应用范围广泛等优点,在实际应用中为众多消息队列中间件产品所采用<sup>[4-7]</sup>。共享内存的分配策略有两种:静态分配方式和动态分配方式,其中静态分配策略应用最为广泛。因为消息队列中间件系统采用共享内存技术来组织消息队列可以不同程度地提高系统的性能,所以研究和探讨共享内存式消息队列与消息分发器技术及其相互关系有很多理论和现实意义。

### 3 消息队列中间件中消息队列与消息分发技术

根据实际系统的需要,消息队列中间件中消息队列和消息分发器的对应关系主要有单消息分发器单队列、单消息分发器多队列、多消息分发器单队列、多消息分发器多队列和二级队列等方式。

#### 3.1 单消息分发器单队列方式(SDSQ)

单消息分发器单队列方式(Single Dispatcher Single Queue, SDSQ)是中间件系统只有一个消息分发器,所有消息都写入同一个共享内存式消息队列当中。其逻辑图如图3所示。

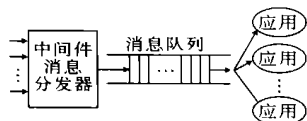


图3 SDSQ结构示意图

这种方式下,客户的所有请求消息都由单一的消息分发器写入同一消息队列中,应用程序的消息处理进程则从消息队列中取出消息作相应处理。SDSQ的优点是队列管理简单,使用方便;但其缺点是无法区分不同的消息类型和不同的消息请求优先级,因而无法处理紧急请求和突发事件。另外,当客户请求增加时,如果应用程序的处理速度小于消息分发器的分发速度,可能出现队满的情况,从而导致消息分发器丢弃消息。

#### 3.2 单消息分发器多队列方式(SDMQ)

单消息分发器多队列方式(Single Dispatcher Multi-Queue, SDMQ)的基本思想是,消息队列中间件系统使用一个消息分发器将客户的所有请求消息或者按分类或者按优先级写入不同的消息队列中,每一队列由消息处理应用程序分别处理。其逻辑图如图4所示。

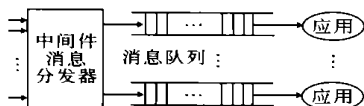


图4 SDMQ结构示意图

SDMQ的核心技术是要求应用程序与消息分发器之间必须制定好通讯协议,当消息分发器接收到消息后,

能方便地按照协议从接收到的信息中取出与消息队列相关的信息,从而将消息写入对应的消息队列中。SDMQ的优点是特别适合需要提供多服务、多优先级的消息队列中间件系统,因而在众多消息队列中间件产品中得到了广泛的使用;其缺点是队列管理比较复杂,实现难度较大,确定消息应当写入的队列有一定的时间开销。此外,如果请求类型(或者优先级)太多,消息分发器可能成为系统的瓶颈。

#### 3.3 多消息分发器单队列方式(MDSQ)

多消息分发器单队列方式(Multi-Dispatcher Single Queue, MDSQ)的基本原理是,多个消息分发器可以同时接收相同或者不同客户的消息请求,但将这些消息写入同一消息队列。MDSQ的逻辑图如图5所示。

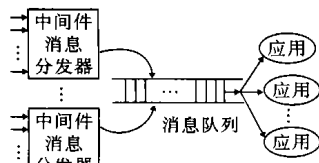


图5 MDSQ结构示意图

MDSQ方式没有解决SDSQ的问题,反而提高了开发的难度,因而在实际系统中很少使用。但如果需要提供多种网络服务(如通过不同的端口发送和接收数据),而每一服务的信息量很小,则可以考虑使用这种技术来设计和开发专用消息队列中间件。

#### 3.4 多消息分发器多队列方式(MDMQ)

多消息分发器多队列方式(Multi-Dispatcher Multi-Queue, MDMQ)是综合了MDSQ和SDMQ的优点,充分利用消息分发器协作关系的一种管理技术。其逻辑图如图6所示。

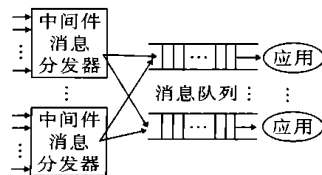


图6 MDMQ结构示意图

MDMQ方式由于存在多个消息分发器,每一个消息分发器只提供一种服务,从而提高了系统的效率;同时,又存在多个消息队列,可以很好地解决请求的优先级问题。MDMQ的最大问题是当多个消息分发器访问同一个消息队列时,可能出现访问冲突。这是因为同一个消息分发器可能将消息写到不同的队列中(如不同的优先级),而不同的消息分发器也可能将消息写到相同的队列中(如相同的服务类型)。利用操作系统提供的互斥功能(如Mutex, Semaphore等技术)可以很好地防止访问冲突<sup>[9]</sup>。但这种解决方法降低了资源的利用率,而且使得消息队列的数量增加,为软件设计增加了难度,系统的效率受到一定程度的影响。

#### 3.5 二级队列方式(SLQ)

二级队列方式(Second Level Queue, SLQ)主要针对MDMQ中的访问冲突提出来的,它采用主从消息分发器和主从消息队列实现消息队列管理。其逻辑图如图7所示。

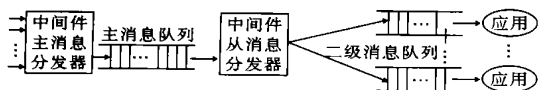


图 7 SLQ 结构示意图

在 SLQ 方式中, 消息分发器收到客户请求消息后, 按入队的方式将该消息写入主消息队列中, 而二级消息分发器或者按服务类型或者按优先级再依次将消息分别写入不同的二级消息队列中。SLQ 方式最大优点是主消息分发器执行效率非常高, 可以解决多服务、多优先级和访问冲突等问题。但在这种方式中, 要求第二级消息分发器执行速度必须与主消息分发器协调, 否则可能导致主消息队列满的情况。一般而言, 服务类型增加, 出现主消息队列满的可能性增加。

4 消息队列与消息分发器组合关系的队列利用率比较

消息队列与消息分发器之间的各种组合关系对消息队列中间件的性能、效率、可靠性等都有很大的影响。表 1 是它们的比较情况。

表 1 消息队列与消息分发器之间各种组合关系比较

内存管理方式	消息分发器效率	能否提供多服务	能否提供多优先级	实现难易程度	应用系统的范围
SDSQ	高	不能	不能	容易	小型专用
SDMQ	低	能	能	容易	中型
MDSQ	较低	不能	不能	容易	中小型
MDMQ	高	能	能	难	大型
SLQ	高	能	能	难	大型

从表 1 中比较可以发现, 单队列方式都不能提供多服务、多优先级的功能, 只能在中小型中间件系统中使用; 多消息分发器和多队列方式均可提供多服务类型、多优先级的功能, 消息分发器执行效率也较高(多分发器单队列 MDSQ 方式除外); MDSQ 虽然可以提供多服务类型、多优先级的功能, 但由于存在资源互斥, 消息分发器的效率反而降低; SDSQ, SDMQ, MDSQ 队列管理容易, 而 MDMQ 和 SLQ 软件实现比较复杂。

(上接第 28 页)的具有最大绝对值的特征根的特征向量。 $x$  即为相关主题的由各个 Web 页的权威值组成的权威值向量,  $Y$  即为相关主题的由各个 Web 页的目录值组成的目录值向量。

3 结论

在巨大并日益增长的 Web 信息资源中搜索相关主题的最具权威的 Web 页有着重要的意义。本文提出了一种计算 Web 页的权威值的算法。该算法改进了 HTIS 算法, HTIS 采用的基于内容的关键词查询获得相关主题的例子页集, 用户需提供相关主题的关键词。本文提出的改进算法无须用户提供关键词, 仅仅采用相关主题的 Web 例子页的连接扩展获得该 Web 页的相关主题的例子页集, 由该例子页集构造基于链接的链接图, 再在链接图中经过重复副本消除合并以及赋予链接图的边初始权值得到带权有向链接图, 由带权有向链接图的连接矩阵基于一个 Web 页被超链接引用的次数来循环更新计算权威向量和目录向量, 在矩阵理论的基础上, 得到

5 总结

消息队列中间件中消息队列和消息分发技术具有重要的地位, 并且与消息队列密切相关。在消息队列中间件系统中, 常用的消息分发器与消息队列技术有 SDSQ, SDMQ, MDSQ, MDMQ 和 SLQ 方式。其中 MDMQ 和 SLQ 应用最广泛, 但技术难度也最大; 其它方式在具体的应用场合中也有很好的应用价值。

参考文献:

[ 1 ] 周世杰. 网络环境下消息队列中间件设计与开发[ D] . 成都: 电子科技大学硕士论文, 2001.  
[ 2 ] 周世杰, 秦志光, 刘锦德. 消息队列技术研究: 综述与一个实例[ J] . 计算机科学, 2002, 29(2): 84-86.  
[ 3 ] Andrew S Tanenbaum. 操作系统: 设计与实现(第 2 版)[ M] . 北京: 电子工业出版社, 1998  
[ 4 ] 刘祖亮. UNIX 技术——系统程序设计篇[ M] . 北京: 电子工业出版社, 2000  
[ 5 ] Andrew S Tanenbaum. Computer Networks[ M] . 北京: 清华大学出版社, 1998  
[ 6 ] Rebert L Knuse. Data Structure & Program Design in C[ M] . 北京: 清华大学出版社, 1999  
[ 7 ] William Stallings. Operating System Internals and Design Principle[ M] . 北京: 清华大学出版社, 1998.  
[ 8 ] W Richard Stevens. UNIX Network Programming, Volume 1[ M] . 北京: 清华大学出版社, 1998  
[ 9 ] 苏森, 唐雪飞, 刘锦德. 面向对象互操作技术[ J] . 电子科技大学学报, 1998, 27(1): 90-94  
[ 10 ] Rhys Lewis. Advanced Message Application with MSMQ and MQSeries[ J] . QUE, 1999, (11).

作者简介:

陆庆(1965-), 女, 工程师, 大学本科, 主要研究方向为计算机网络与中间件技术; 周世杰, 男, 博士研究生, 主要研究方向为开放系统与中间件技术、ITS 技术; 秦志光, 男, 教授, 博士, 主要研究方向为开放系统与安全技术、ITS 技术; 刘锦德, 男, 教授, 博士生导师, 主要研究方向为开放系统与中间件技术、网络与多媒体技术。

了权威向量和目录向量的计算公式。该算法可用于实际的搜索引擎系统中。

参考文献:

[ 1 ] J Kleinberg. Authoritative Sources in a Hyperlinked Environment[ C] . Proc. 9th ACM-SIAM symposium on Discrete Algorithms, ACM Press, 1998 668-677.  
[ 2 ] S Chakrabarti, et al. Automatic Resource Compilation by Analyzing Hyperlink structure and Associated Text[ C] . Proceedings of the 7th World-Wide Web conference, 1998 65-74.  
[ 3 ] Chia-Hui Chang, Ching-Chi Hsu, Enabling Concept-based Relevance Feedback for information Retrieval on the WWW[ J] . IEEE Transaction on Knowledge and Data Engineering, 1999, 11(4): 595-609.

作者简介:

汤艳红(-), 男, 硕士研究生, 主要研究方向为数据挖掘和信息安全; 李石君(-), 男, 博士, 副教授, 主要研究方向为数据挖掘和信息安全; 黄竞伟(-), 男, 博士, 教授, 博士生导师, 主要研究方向为演化算法。