文章编号:1008-0570(2006)07-3-0064-03

基于 EPOLL 机制的 LINUX 网络游戏服务器实现方法

Realization of EPOLL-based Linux Online Games Server

(1.西安电子科技大学;2.上海大学) 崔　滨[1]　万旺根[2]　余小清[2]　楼顺天[1]
Cui,Bin　Wang,Wangen　Yu,Xiaoqing　Lou,Shuntian

摘要: 文章论述了在 Linux 平台上一种高效的 I/O 方法—epoll, 针对网络游戏中大量并发客户请求问题,提出采用 epoll 机制建立高效网络游戏服务器思想, 较好地解决了网络游戏服务器中的大量用户并发接入问题。 同时在服务器整体设计的基础上,给出了在 Linux 环境下基于 epoll 机制的网络游戏服务器实现方法。

关键词: Linux 操作系统; epoll 机制 ;网络游戏
中图分类号: TP311.52　　**文献标识码**:A

技术创新

Abstract: This paper introduces an efficient Input/Output method—epoll, based on Linux platform. Aiming at a plenty of clients requests in Multiplayer Online Games ,this paper recommends an idea of building effective Game server which uses epoll. The method solves the problem of a great deal of clients requests successfully. After designing the whole structure of the Game server, the author gives the realization of epoll mechanism in Multiplayer Online Game server in Linux environment.

key words: Linux ,epoll , multiplayer online game

1　引言

随着网络游戏产业在中国的迅猛发展,游戏玩家人数不断增多, 玩家对游戏品质的要求也不断提高, 运营商需要更加高效、稳定、安全的网络游戏服务器。采用 Linux 架构的网络服务器无论在安全性、可靠性还是可升级性方面都具有优势, 另外 Linux 架构成本更低, 比微软或 Unix 操作系统平均低出 1/3 左右,所以有较强的竞争力。然而在 2.5 内核之前,Linux 系统在网络通信中使用的多线程同步阻塞 I/O , select/poll 单线程事件驱动 I/O 等技术都有一定的缺点, 在解决大量用户并发请求问题上不如微软 windows 的 IOCP 机制。不过随着最近 Linux2.6 内核的推出,其中的 e-poll 机制为解决大量并发用户请求问题提供了一个良好的解决方案。epoll 机制用于网络游戏的通讯层,结合游戏协议层和基于线程池的逻辑处理层可构造一个高效的网络游戏服务器。

2　epoll 机制

Linux2.6 内核中集成了新的单线程事件驱动 I/O 技术—epoll, 传统的 Linux 处理大量用户连接主要使用二类方法,一是多进程或多线程同步阻塞 I/O 技术,二是 select/poll 的单线程事件驱动 I/O 方式。多线程或多进程同步阻塞网络 I/O 技术处理高并发的网络连接,虽然直观方便,但要求给每个用户开一个连接线程,首先创建线程以及上下文切换操作时会产生较大的系统开销,二是内存开销较大,所以不能满足大量用户的请求。select/poll 的单线程事件驱动 I/O 方式是经典的 Unix/Linux 处理 I/O 技术,这两种技术的原理是在多个文件描述符上等待 I/O 事件,与多进程或多线程同步阻塞 I/O 技术相比虽然没有创建大量线程和上下文切换等开销, 但系统内核必须遍历整个文件描述符集来判断是否有非阻塞的 I/O 就绪,另外还要调用进程等待那些可能阻塞的描述符队列, 在接入用户不是很多的情况下 select/poll 还算高效, 但当进程管理成千上万个 socket 文件描述符时,采用这两种技术的服务器伸缩性不好,不能满足大量用户的并发请求。

2.1 EPOLL 机制概述

以 Davide Libenzi 为首的 linux 开发人员经过长时间工作,创造了一种新的管理大量文件描述符的技术,最早是通过一特殊设备/dev/epoll 来实现的。后来应 Linux 创始人 Linus 的要求,转变成了一组 epoll()系统调用。 epoll 有两种工作方式 epoll-LT 和 epoll-ET:

epoll-LT(level triggered)是缺省的工作方式,同时支持阻塞和非阻塞 socket 工作模式。在这种工作方式下,如果一个文件描述符就绪,内核通知应用程序对该描述符进行 IO 操作,如果不作任何操作,内核还会继续通知应用程序。

epoll-ET(edge-triggered)是高速工作方式,只支持非阻塞 socket 工作模式。在该工作方式下,当描述符

崔滨:硕士生
基金项目:上海市科委重大科技攻关项目(NO.045115014)

从未就绪变为就绪时，内核通过 epoll 通知应用程序。但不会再为那个文件描述符发送更多的就绪通知，直到执行某些操作导致那个文件描述符不再为就绪状态了。

2.2 EPOLL 机制的优点

epoll 调用与 Unix/Linux 使用的经典单线程事件驱动 I/O 方式相比有如下优点：

(1) 支持进程打开巨量文件描述符

select 最大的缺点是一个进程所打开的文件描述符是有一定限制的，默认值是 2048。对于那些需要支持的成千上万连接数目的服务器来说显然太少了。e-poll 则没有这个限制，它所支持的文件描述符上限是最大可以打开文件的数目，具体数目和系统内存大小有关。如在 1GB 内存的服务器上可打开的文件描述符数大约是 10 万左右。

(2) IO 效率不随文件描述符数量增加而线性下降

select/poll 即使在任一时间内只有部分的 socket 描述符“活跃”，每次调用还会扫描全部的集合，这样在描述符集合很大的情况下，会导致效率呈线性下降。但是 epoll 不存在这个问题，它只会对“活跃”的 socket 进行操作。所以在只有部分描述符“活跃”的环境中，epoll 的效率远在 select/poll 之上。

(3) 加速内核与用户空间的消息传递

无论是 select,poll 还是 epoll 都需要内核把文件描述符上发生的消息通知给用户空间，如何避免不必要的内存拷贝就很重要，epoll 是通过内核在用户空间映射同一块内存实现的，加速了内核与用户空间的消息传递。

(4) epoll 拥有稳定的数据结构使系统具有较好的伸缩性

select/poll 每次调用的是独立的事件，在每次调用之后 FD_SET 或 POLLFD 的结构内容会被清除，需要重新设置。相反，epoll 要求应用程序首先在内核空间中建立一个稳定的数据结构 epoll_event 用于注册所感兴趣的事件和回传所发生待处理的事件，增强了系统伸缩性。

3 网络游戏服务器的实现

网络游戏服务器与其他应用系统服务器如 web 服务器，企业应用服务器等相比有自己的特点，一是管理巨量客户连接，二是客户连接上线后会一直保持与服务器的连接，三是每个玩家的 socket 不是每时每刻都是活跃的。比较适合发挥 epoll 的优点为大量并发用户提供即时服务。

网络游戏服务器的结构可分为四个层次，如图 1 所示，其中网络通信层负责处理游戏中玩家与服务器之间的网络传输细节，实现数据快速有效的传输。在这里可使用 EPOLL 机制实现高效的网络通信。游戏协议层负责处理游戏中玩家与服务器之间交互传递的数据包，游戏逻辑层负责处理游戏逻辑，包括大厅模块（包括房间子模块、交易子模块）、用户管理模块等。数据库层的主要功能是完成数据库系统的连接、实际数据操作等功能，提供物理无关的数据库操作接口。

3.1 epoll 机制在通信层的应用

epoll 机制主要负责对大量并发用户的请求进行及时处理，完成网络游戏服务器与客户端的数据交互，如图 2 所示，其具体实现步骤如下：

(1) 使用 epll_create()函数创建 epoll 文件描述符，设定将可管理的最大 socket 描述符数目，根据不同性能的服务器，设定不同的值，一般可设为 500-1000。

(2)创建与 epoll 关联的接收线程，一个应用程序可以创建多个接收线程来处理 epoll 上的读通知事件，线程的数量依赖于程序的具体需要。考虑到网络游戏中的实际需要，在 3D 场景中玩家数据包会很频繁地到达服务器端，所以接收线程个数最好等于中央处理器的个数，以便充分发挥系统的潜力。

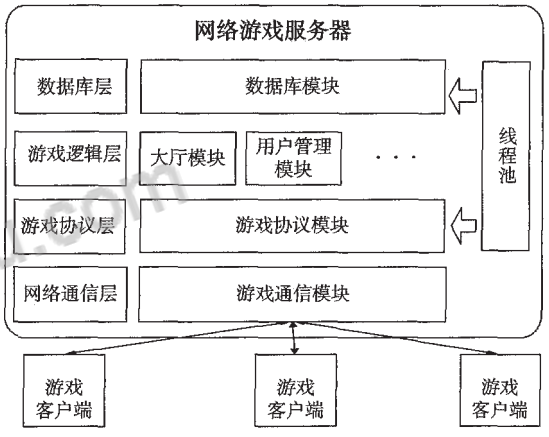

图 1 网络游戏服务器层次结构

Fig. 1 the structure of Online Game Server

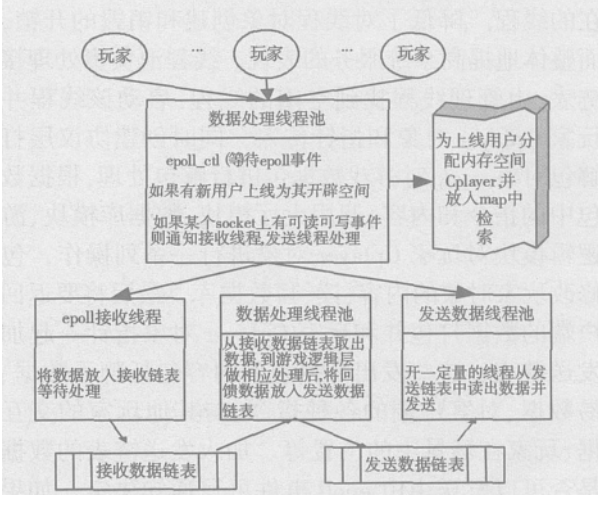

图 2 EPOLL 的应用

Fig.2 The application of EPOLL

(3)创建一个侦听 socket 描述符 Listensock；将该描述符设定为非阻塞模式，调用 listen()函数在该套接字上侦听有无新用户上线，在 epoll_event 结构中设置要处理的事件类型为 EPOLLIN,工作方式为 epoll- ET，以

提高工作效率,同时使用 epoll_ctl()注册事件,最后启动网络监视线程。

(4)网络监视线程启动循环,epoll_wait()等待 epoll 事件发生。

(5)如果 epoll 事件表明有新用户上线,则调用 accept()函数,将用户 socket 描述符加入 epoll_data 联合体,同时设定该描述符为非阻塞,并在 epoll_event 结构中设置要处理的事件类型为读和写,工作方式为 epoll- ET, 另外为新用户创建一个 Cplayer 对象,并以 socket 描述符为关键字加入玩家 map。

(6)如果 epoll 事件表明 socket 描述符上有数据可读,则将该 socket 描述符加入可读队列,通知接收线程读入数据, 接收线程根据 socket 描述符在玩家 map 内找到相应的 Cpalyer 对象,将其放入接收链表,交由协议层和逻辑层处理。

3.2 游戏服务器协议层、逻辑处理层、数据库层的实现

epoll 的接收线程将收到的数据放入接收链表,再交由游戏数据处理线程池进行处理。在网络游戏中用户接入量较大,同时有些对数据包的操作比较耗费服务器资源,例如数据包解密、对数据库的读写操作以及一些复杂的逻辑处理等。这样就存在大量用户需要服务与服务器资源有限的矛盾,线程池提供了一个解决方案,线程池在程序开始时就在内存中开辟一些固定数目的线程,当有新的客户请求到达时,不是新创建一个线程为其服务, 而是从线程池中选择一个空闲的线程为新的客户请求服务,服务完毕后,线程进入空闲线程池中;如果没有线程空闲的话,就等待线程池内有线程空闲以后再进行处理。通过对多个任务重用已经存在的线程, 降低了对线程对象创建和销毁的开销,从而整体地提高系统服务的表现。线程池负责处理接收链表,由管理线程找到空闲的线程,启动该线程并将玩家 Cplayer 对象和指针传入, 同时创建协议层打包解包对象 pack,对游戏数据包进行解包处理,根据数据包中的指令和内容,调用大厅模块、数据库模块、游戏逻辑模块对玩家 Cplayer 对象进行一系列操作, 包括修改玩家对象的内容,读、写数据库。最后将要返回客户端的数据打包并和玩家 Cplayer 对象指针一起加入发送链表, 这些发出数据包的内容包括聊天数据、交易数据、对客户端的各种指令、与其他玩家的交互数据、玩家在场景中的位置等。加入发送链表的数据包是否可以发送由由 epoll 事件可写通知决定, 如果 epoll 事件通知可写,则设定玩家 Cplayer 对象的发送标志位为 1, 发送线程根据此标志位发送该玩家的数据包。由于 socket 描述符设定的是非阻塞方式,所以在调用发送操作以后, 立即将玩家 Cplayer 对象的发送标志位清 0,以防该次数据还未发完,而下一次的数据又要写入发送缓冲而出现错误,这样直到 epoll 事件通知再次可写后, 下一次的数据包才能发送。如果玩家 Cplayer 对象的发送标志位为 0,表示该用户还未发送就绪,发送线程先发送其他数据包,直到该用户的数据包可发送为止。

4 结语

epoll 能够管理大量的文件描述符。epoll 建立的稳定的数据结构增强了系统伸缩性,并且不用轮询集合中的所有描述符来判断是否有描述符就绪,大大提高了处理大量并发用户请求的能力,因此通过 epoll 管理 socket 描述符的效率要明显高于 linux 架构下的其他的 I/O 模型, 读线程能够最及时地得到 epoll 的通知,比较适合网络游戏服务器大量用户并发接入的特点。另外由于整个网络通信过程均是采用事件驱动 I/O 方式, 所以网络游戏服务器能够同时执行读/ 写操作,异步处理大量客户请求, 并且最大限度地提高了系统的性能和速度。编写网络游戏服务器程序的难点在于程序的“可扩展性”,即如何开发出大容量且能处理大量并发套接字 I/O 请求的应用程序。epoll 机制是一种与 linux 结合度较高的实现高效率 I/O 的有效方法。因此在 Linux 操作系统下构建网络游戏服务器时, 应优先考虑 epoll 以获得更好的性能和强大的扩展能力。

参考文献:

[1]乔晓丹,张鹏.一个基于 Linux 操作系统的嵌入式网关的实现[J].微计算机信息,2005,7- 2:26- 28

[2]sys_epoll - making poll fast [EB/OL]. http://lwn.net/Articles/14168/,2005

[3]滕昱.Linux2.6 内核 epoll 介绍[EB/OL]. http://www.blogdriver.com/mechgouki/601157.html,2005

[4] Louay Gammo, Tim Brecht, Amol Shukla, David Pariag.Comparing and Evaluating epoll, select, and poll Event Mechanisms [EB/ OL] .http://gelato.uwaterloo.ca,2005

[5]苏羽,王媛媛. 网络游戏建模与实现[M]. 北京:北京科海电子出版社,2003

[6]李昊,刘志敬.线程池技术的研究[J]. 现代电子技术 2004 3: 80

作者简介: 崔滨(1974~)男,吉林省长春市人, 硕士生, 主要从事网络游戏引擎开发方面的研究

(710071 陕西 西安电子科技大学电子工程学院) 崔滨 楼顺天

(200072 上海 上海大学通信与信息工程学院)万旺根 余小清

(School of Electronic Engineering , Xidian University, Shannxi 710071, China) Cui,Bin Lou,Shuntian

(School of Communication and Information Engineering, Shanghai University, Shanghai 200072, China)Wang,Wangen Yu,Xiaoqing

通讯地址: (200072 上海市延长路 149 号上海大学延长校区西区宿舍 C4—306) 崔滨

(投稿日期:2005.12.16) (修稿日期:2006.1.24)