

分布式实时游戏服务器集群架构设计

许登元,王 鹏

(重庆交通大学 信息科学与工程学院, 重庆 40074)

摘 要: 为保证网络游戏的通畅性,设计一种具有高性能的分布式游戏服务器架构.通过精简通信协议、数据包过滤机制从微观上减少单个数据包处理时间;利用集群技术构建服务器组从宏观上提高服务器处理能力.在此基础上构建出一个具有高性能、分布式、实时性较强的游戏服务器集群,并对该架构的处理时间性能进行了测试与分析.测试表明,在十万数量级并发请求情况下,分布式服务器集群比单个服务器处理时间性能提高 1.8 倍.

关键词: 分布式系统;网络服务器;通信协议;集群

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2014)07-0105-04

The Design of Distributed and Real-time Game Server Cluster Architecture

XU Deng-yuan, WANG Peng

(School of Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China)

Abstract: A kind of web servers' architecture with high performance was designed in this paper in order to ensure the fluidity of network game. The processing time of packets was reduced by simplified communication protocol and packet filter mechanism. Servers' processing ability can be improved by building server clusters. Then we design a distributed and real-time web servers' cluster with high performance. The time performance of the web servers' cluster was test and analyzed at last. Tests show that the distributed server clusters can improve time performance about 1.8 times compared to a single server in one hundred thousand concurrent requests

Key words: distributed system; network server; communication protocol; cluster

1 引言

近年来,伴随着计算机和网络的日益普及,游戏行业取得了长足的发展.游戏行业从最早期的单机类型逐步发展成为大型多玩家在线游戏(Massive Multiplayer Online Games, MMOG)^[1].随着网络带宽的增加,愈来愈多的人选择网络游戏来作为一种新型娱乐方式^[2].因此,提供良好的服务性能,保证网络游戏的实时性、顺畅性是考验一款网络游戏的关键环节.而游戏服务器集群作为整个游戏的中心,负责整个游戏的运行.对此,本文提出一种分布式实时性的游戏服务器集群架构.

2 架构总体设计

良好的结构设计会减少游戏服务器的处理时

间,本文所设计的游戏服务器集群基础架构如图 1 所示

整个架构分为 5 个子系统:网关服务器(GateServer)、消息路由服务器(NetMessageRouter)、登陆服务器(LoginServer)、游戏服务器(GameServer)、数据库服务器(DbServer).

该游戏服务器集群基础架构具有如下特点:

①使用 GateServer 隔离客户端与服务器,最大程度上减少客户端对服务器的直接访问,减少非法的数据访问,从而提高服务器的安全性^[3].

②在玩家发出请求后,所有请求均需要经过消息路由服务器,因此,消息路由服务器的包过滤功能可以提前处理非法或错误的包,从而节省其他服务器的处理时间.

③各个服务器之间相互独立,服务器之间的通

收稿日期: 2013-10-10; 修回日期: 2013-12-01

基金项目: 重庆市自然科学基金项目(cstc2013jcyjA30016)

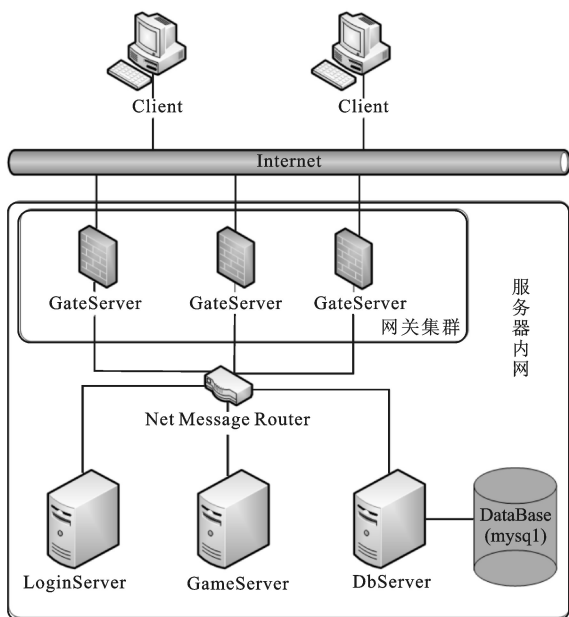


图 1 游戏服务器集群基础架构图

信采用统一的协议,保证服务器之间的连通性。

④良好的扩展性和伸缩性. 对于小型的网络游戏,可以将几个服务器放置同一个物理服务器上,以便节省资源,减少成本. 对于大型的网络游戏,可以将各个服务器程序放置在不同的物理服务器上,增强整个服务器集群的处理能力,从而提高游戏响应速度。

2.1 网关服务器

网关服务器主要负责客户端与服务器之间的连接,所有的客户端统一连接到网关服务器. 再由网关服务器将数据包转发给消息路由服务器进行二次转发. 因此,网关服务器很好地隔离了客户端与服务器的连接. 对于客户端来说,服务器是透明的. 反之,对于服务器来说,客户端也是透明的。

一般来讲,为了提高整个游戏的响应时间,网关服务器会进行一些简单的过滤。

2.2 消息路由服务器

由于各个服务器之间相互独立,它们之间的通信是通过统一的通信协议来进行. 因此,消息路由服务器需要负责各个服务器之间的路由服务. 与此同时,它还负责对数据包的完整性和合法性进行校验,通过校验则进行转发. 否则,将抛弃该数据包。

2.3 登陆服务器

主要负责玩家账号的验证和本次 SessionKey 的生成(所谓 SessionKey, 是客户端本次登陆后,服务器生成的通信密钥). 在生成 SessionKey 之后,登陆服务器会将其发送给各个服务器和客户端,在客户端后续请求的数据包中均含有该 SessionKey,服

务器将其作为玩家的登陆凭证,进而进行数据验证。

2.4 游戏服务器

游戏服务器是整个架构的中心,它主要服务游戏业务逻辑相关的请求处理. 通俗来讲,它包含游戏的各个子系统或模块. 例如:角色属性系统、宠物系统、战斗系统、场景系统等。

玩家在登陆之后,玩家的请求经由消息路由服务器转发至游戏服务器处理. 处理完毕后,将请求的数据返还给网关服务器,进而反馈到客户端。

2.5 数据库服务器

数据库服务器主要负责需要持久化数据的保存. 对于服务器而言,整个系统中每个玩家获得的属性不尽相同. 因此需要对相关的数据进行保存,以便维护用户数据的连续性。

3 通信协议设计

对于游戏响应速度而言,尽量减少占用带宽可以很大程度上提升游戏实时性. 因此,设计一种精简合理的通信协议可以减少对带宽的占用,提高游戏响应速度。

因此本文设计的通信协议如表 1 所示。

3.1 协议结构设计

表 1 游戏服务器集群采用的通信协议

消息序号	消息长度	会话密钥	消息类型	消息内容	校验码
msgSeq	msgLength	sessionKey	msgType	Msg	checkSum

在该结构中,消息序号(msgSeq)是请求的序列号,其生命周期贯穿于整个请求周期. 从发起请求,到服务器处理完毕,最后返回到客户端,均将其作为关键标志. 首次登陆后初始化 msgSeq 为 0,之后一直递增,直至玩家下线。

消息长度(msgLength)是用来标示该数据包的长度. 由于本文采用的消息载体 ProtoBuffer 没有自带长度信息或终结符,因此需要应用程序在发送和接受的时候自己来做正确的切分. 所以,协议中应该携带有消息长度信息^[4]。

会话密钥(SessionKey)是由客户端首次登陆之后,服务器生成的唯一标志. 其生命周期贯穿于整个游戏服务期. 对于本文而言,采用 MD5 散列的方式来生成 SessionKey^[5]。

$$\text{SessionKey} = \text{MD5}(\text{roleId} + \text{salt}) \quad (1)$$

式中,roleId 是玩家的标志 ID, salt 是本次登陆随机产生的 5 位整数. SessionKey 是散列之后的字符串。

消息类型(msgType)是用来标示该消息的请求目标.具体而言,本文中的消息类型可取值范围(客户端(0x01),网关服务器(0x02),游戏服务器(0x03),数据库服务器(0x04),登陆服务器(0x05)).消息路由服务器会对类型进行检查,进而将携带的数据转发给对应的服务器.

消息内容(Message)是协议携带的具体数据.在本文中是 ProtoBuffer 序列化之后的二进制数据流.

校验码(CheckSum)是用来维护整条消息数据的有效性,保证在传输过程中没有产生错误.其产生如式(2)所示.

$$\text{Checksum} = \text{MsgSeq} + \text{MsgType} + \text{SessionKey} \quad (2)$$

3.2 协议处理流程

对于客户端发起的请求,在将数据包发送到网关服务器之后,网关服务器会提取校验码和消息类型.首先利用服务器段保存的 SessionKey 和 msgSeq 进行校验码验证.验证未通过,则断开客户端与网关服务器之间的连接.反之,将数据包进一步分发给消息路由服务器进行处理^[6].

对于服务器端返回数据包,网关服务器也会进行校验码进行验证.验证未通过,则直接抛弃该数据包.反之,将数据包返还给客户端.

4 消息载体和消息队列

消息是数据包的核心内容,包含具体业务请求数据.在本文架构中,ProtoBuffer 被用作承载具体的数据包内容.

ProtoBuffer 定义了一种紧凑的可扩展二进制消息格式,特别适合网络数据的传输.同时它提供了多种语言的使用接口,大大方便了分布式程序的开发^[7].

对于网络请求而言,大多具有突发性.因此,在网络较为拥堵或超过服务器处理能力的时候,如何处理大量的请求成为考验服务器性能的关键点.除了采取常规提高服务器的物理性能之外,设计优良的消息队列可以很程度上提高处理能力.

ZeroMq 是一种基于消息队列的多线程网络库,其对套接字类型、连接处理、帧,甚至路由的底层细节进行抽象,提供跨越多种传输协议的套接字. ZeroMq 是网络通信中新的一层,介于应用层和传输层之间,是一个可伸缩层,可并行运行,分散在分布式系统间^[8].

5 对象池和负载均衡

游戏服务器要求的实时性远远高于一般的 Web 服务(Web 服务的工作流程图如图 2 所示).由于 Web 服务工作流对于每次工作请求都会转发给 DB 处理,而 DB 处理相对于内存处理来讲速度较慢.因此,Web 服务中常见的 Request → DbServer → Respond 模式并不适用于游戏服务器.

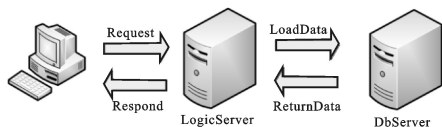


图2 Web服务工作流

内存数据库是新近出现的一种数据库类型.其核心是将需要处理的数据存放在内存之中^[9].本文未采用专门的内存数据库,而是采用其核心思想利用对象池技术来满足自己的需求:在玩家首次登陆阶段就开始拉取玩家相关的数据,将其保存在内存的对象池中.由于内存查询速度远远大于数据库查询速度,因此可大大缩短处理时间.在玩家发出下线请求之前,游戏服务器进行玩家相关数据的内存清理,将玩家数据同步到数据库中,游戏玩家上线的处理流程如图 3 所示.

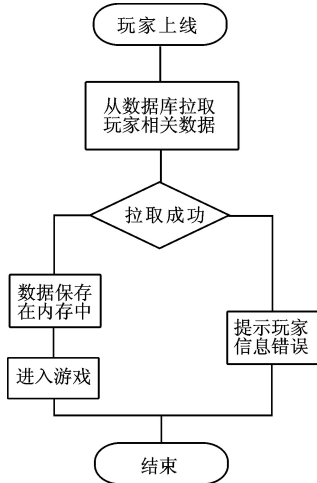


图3 玩家上线流程

与此同时,多人在线的网络游戏往往伴有大量并发请求,如果对于每一个请求都进行生成请求、处理请求、答复请求的流程,在内存空间、对象生成时间上会有较大的浪费.因此,本文在游戏服务器初始化阶段就生成一个请求池(RequestPool).对于以后游戏过程中的请求,只需要到 RequestPool 中生成一个请求实例即可,从而减少内存资源和生成对象时间上的浪费,玩家请求处理流程图如图 4 所示.



图4 玩家请求处理流程图

精简的通信协议设计、快速的消息队列、对象池技术都能很大程度上提高游戏服务器处理性能。但是这些都是从微观上保证性能。从宏观而言，面对上百万同时在线玩家的请求，这些措施远远不够。因此，采用集群技术显得尤为重要。

负载均衡即为集群技术的一种。如前文所言，整个游戏请求都会经过网关服务器，这样带给网关服务器很大的压力。同时，网关服务器的处理能力将成为影响服务器系统性能的瓶颈。因此，在用户首次登陆阶段，就采用负载均衡手段，将用户请求分布到多个网关服务器上进行处理。这种多服务器的技术很大程度上可缓解大量玩家的请求阻塞情况，即为分布式处理。

6 架构性能测试及分析

本文采用服务器硬件指标如表2所示。在此基础上，模拟客户端请求，测试架构在不同数量级并发请求的处理时间，测试结果如图5所示，采用均衡负载技术后的处理时间如图6所示。

表2 服务器硬件指标

操作系统	CPU	内存	带宽
CentOS6.3	IntelE5345@2.33 GHz	16 GB	100 Mb/s

由图5可以看出，单个服务器的处理时间伴随着数量级的增加而增加，在数量级达到十万级时服务器的处理能力达到瓶颈。当同时在线人数达到百万级时，单个处理请求需要将近800 ms。此时的请求基本处于阻塞状态。

针对单个服务器并发处理能力不足的问题，当同时在线人数达到十万级时，采用负载均衡技术（即增加服务器集群），可有效降低处理时间。从图6可

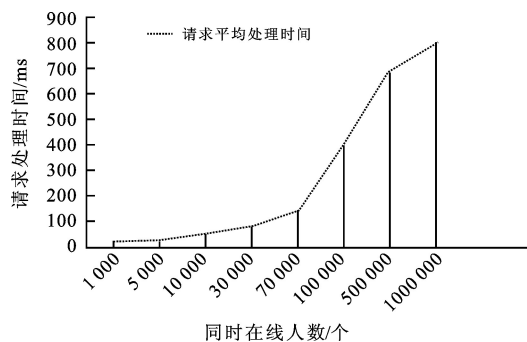


图5 单个服务器不同数量级请求处理时间

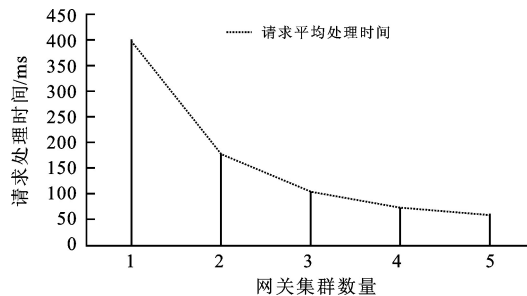


图6 10万在线人数下，不同集群数请求处理时间

以看出，在十万数量级并发请求情况下，服务器个数由1增加到3时，处理时间降低290 ms左右，极大的提高处理时间。

7 结束语

为满足游戏的实时性和高并发的特性，本文着重从微观和宏观上做出优化。精简的通信协议、良好的过滤机制从微观上缩短每个数据包的处理时间。对象池、负载均衡从宏观上保证对请求的实时处理。并在此基础上构建出分布式实时性游戏服务器集群，从架构性能分析可以看出，该架构具有良好的扩展性和伸缩性。

参考文献：

- [1] 吴晓东,李建微,陈新. 基于MMORPG的网络同步问题的研究与实现. 现代计算机, 2011. 9.
- [2] 2012年中国网络游戏市场年度报告[R]. 2013. 5.
- [3] 贺冯政. 网络游戏服务器通信架构及关键技术研究[D]. 四川:电子科技大学, 2008.
- [4] 杨玲. 一种高性能网络服务器架构设计[J]. 网络安全技术与应用, 2010(4): 59-61.
- [5] 孙维国,李浩然. MD5算法在数据安全中的应用及安全性分析[J]. 微计算机应用, 2010(10): 66-69.
- [6] 吕琳,陈永辉,向科峰. 网络游戏中通信模块的设计与分析[J]. 电脑知识与技术, 2006(10): 58-63.
- [7] 王少林,李祥明,李永刚,等. 基于Protocol Buffers的航天测控数据交换模式研究[J]. 遥测遥控, 2012, 33(4): 23-27.

(下转第112页)

所示.

从实验结果来看,本模型具有较高的检测率.不论是在真实的网络数据流还是 KDDCup99 入侵检测评估数据上,均取得了较好的实验结果.本模型的检测率平均可以达到 97% 左右,虚警率可以降低到 4% 以内.与文献[8]相比较,本文的检测率和虚警率都有所降低.相比于文献[9],本模型的检测率有明显提高,但虚警率也有所增加,但是幅度不大(检测率和虚警率是相互制约的),本模型将危险理论和否定选择机制结合使用,提高了检测性能,也验证了思路的可行性.

表 3 Kddcup99 数据集上的性能比较

方法	检测率/%	虚警率/%
文献[8]	94.2	2.51
文献[9]	79.8	1.73
本文算法	96.3	2.44

5 结束语

本文提出了一种集成免疫否定选择机制和危险理论的入侵防御模型.实验结果表明,本模型有较高的检测性能.如何进一步优化危险信号的提取和检测器的生成及入侵防御手段的完善是下一步研究的方向.

参考文献:

- [1] 柴争义,刘芳,朱思峰.新型智能入侵防御模型[J].华中科技大学学报:自然科学版,2010,31(1):28-30.

- [2] 张然,刘洋,尹毅锋,等.基于 Multi-Agent 的入侵检测动态协同机制研究[J].微电子学与计算机,2013,50(8):57-62.
- [3] 傅蓉蓉,郑康锋,芦天亮,等.基于危险理论的无线传感器网络入侵检测模型[J].通信学报,2012,33(9):31-37.
- [4] 陈岳兵,冯超,张权,等.一种面向入侵检测的集成人工免疫系统[J].通信学报,2012,33(2):125-131.
- [5] 王彦涛.基于免疫计算的实时入侵检测系统的设计与实现[D].西安:西安电子科技大学,2012.
- [6] 柴争义,王献荣,王亮.用于异常检测的实值否定选择算法[J].吉林大学学报:工学版,2012,23(1):61-65.
- [7] 柴争义,刘芳.应用危险理论的网络安全风险感知模型[J].北京邮电大学学报学报,2010,33(3):40-43.
- [8] Gonzalez F A, Dasgupta D. Anomaly detection using real-valued negative selection[J]. Genetic Programming and Evolvable Machine, 2003, 4(4): 383-403.
- [9] Gu F, Greensmith J, Aickelin U. Further exploration of the dendritic cell algorithm[C]//International Conference on Artificial Immune Systems. Phuket, Thailand:Springer,2008:142-153.

作者简介:

魏 萌 女,(1981-),硕士,讲师.研究方向为网络与信息安全,人工智能,数据挖掘.

贾遂民 男,(1968-),副教授.研究方向为网络异常检测、智能计算.

胡明生 男,(1973-),博士,副教授.研究方向为人工智能、数据挖掘等.

(上接第 108 页)

- [8] 蒲凤平,陈建政.基于 ZeroMQ 的分布式系统.电子测试,2012(7):24-29.
- [9] 袁培森.内存数据库的设计与实现[D].南京:南京航空航天大学,2009.

作者简介:

许登元 男,(1971-),博士,教授.研究方向为计算机网络、无线传感器网络.

王 鹏 男,(1988-),硕士研究生.研究方向为计算机网络、嵌入式.