Java在游戏服务器开发中的应用

发表于2015-05-20 15:55| 1839次阅读| 来源CSDN| 13 条评论| 作者王广帅

[特别策划](http://www.csdn.net/tag/%E7%89%B9%E5%88%AB%E7%AD%96%E5%88%92/news)[Java20周年](http://www.csdn.net/tag/java20%E5%91%A8%E5%B9%B4/news)[Java](http://www.csdn.net/tag/java/news)[《程序员》电子刊](http://www.csdn.net/tag/%E3%80%8A%E7%A8%8B%E5%BA%8F%E5%91%98%E3%80%8B%E7%94%B5%E5%AD%90%E5%88%8A/news)

**摘要：**随着游戏市场的兴起，特别是网页游戏、手机游戏的崛起，对游戏开发技术的需求越来越多。触云游戏公司选择使用Java做服务器开发语言，一起来看下Java在触云游戏服务器开发中的应用。

随着游戏市场的兴起，特别是网页游戏、手机游戏的崛起，对游戏开发技术的需求越来越多。网络游戏开发是一个庞大的体系，总体来说是客户端与服务器端。客户端是玩家接触的游戏图像显示端，服务器是处理游戏运行中的各种数据，由于一台服务器要支持众多玩家的请求，所以服务器的性能高低决定了同一个游戏的用户数量。

我们公司选择使用Java做服务器开发语言，主要原因是：1.Java是跨平台的，方便部署；2.Java是安全的高级语言，可以提高开发效率；3.Java是面向对象的，代码可以重用；4.Java的分布式应用。

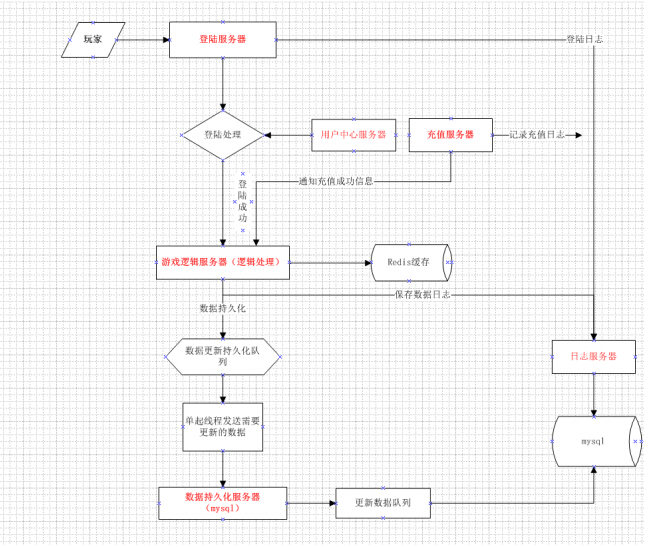


图1 服务器架构图

**服务器架构**

Java在我们的服务器开发中的应用。服务器架构如图1所示。服务器架构的几个模块：

* 登录服务器
* 逻辑服务器
* 用户中心服务器
* 充值服务器
* 数据库服务器
* 日志服务器

这些模块都是分开的，可以灵活地分布式部署到不同的物理服务器上。只需要修改一些配置文件即可，非常方便。

**服务器详细功能模块**

**登录服务器**

负责处理玩家登录的请求。一个登录服务器对应多个游戏逻辑分区。当玩家登录的时候，登录服务器向用户中心服务器发送登录信息。请求对登录信息的验证。通过验证之后，返回分区地址，之后，客户端与登录服务器断开，连接到游戏逻辑服务器。登录服务器是一个单独的Java运行程序，当访问量增加大，可以增加部署到多个物理服务器上面，均衡负载访问压力。它通过使用Java的NIO（非阻塞）方式与客户端进行通信。通过用户中心服务器提供的接口访问用户中心，进行数据处理。

**逻辑服务器**

对玩家的操作进行逻辑处理。逻辑服务器是整个游戏的心脏。它的工作效率直接影响玩家在游戏中的体验，所以对它的要求就是速度，快速返回处理结果。为了达到满足要求的速度，逻辑服务器的大部分操作必须在内存中操作，避免I/O操作，I/O操作可以放到另外的线程中进行。说是大部分，是因为玩家在第一次登录的时候可能会从数据库加载所要用到的数据。在图中，大家看到了缓存，缓存的作用就是把数据放在内存中。当玩家退出时，它的数据也会在缓存中保存一段时间，在一定时间内，玩家再次登录，将不会再重新从数据库加载数据。在逻辑服务器中对数据库的操作可以先放入一个Java队列中，再另起一个Java线程负责从这个队列取数据，并发送到数据库服务器，这就是使用Java的阻塞队列，快速实现一个生产者—消费者模式，数据生产与处理相分离，这样既减轻了逻辑服务器的压力，也保证了数据处理的效率。逻辑服务器的日志也不在逻辑服务器入库，同样的发送到日志服务器处理。还有一种方法是以一种特定格式的方式，记录到本地文件中，再启动一个进程，读取这个文件，然后入库。

**用户中心服务器**

现在很多游戏都对用户进行了集中管理。这方便了对用户提供更好的服务，比如充值、活动、礼包领取、新游戏导入用户等。有的游戏公司可能会用用户中心的数据发展游戏运营平台。这部分与游戏逻辑服务器分开，也减少了游戏逻辑服务器的压力。用户中心采用JavaWeb开发，它对游戏服务器只提供特定访问的接口，把数据与逻辑分离开来，方便管理，以及分布式部署，增强了架构的灵活性。

**充值服务器**

充值是游戏收入的唯一方式，所以这个功能必须流畅，毫无压力。如果由于网络或服务器性能原因，导致玩家充值不了，会直接影响收益的。所以充值服务器最好部署在一台单独的物理机上面，也可以多个分区使用一个充值服务器，这要视游戏人数而定。

**数据库服务器**

负责对数据入库及更新的操作。把这部分操作从逻辑服务器分离出来，就是为了减轻逻辑服务器的压力，减少逻辑服务器资源的占用。而且，如果逻辑服务器突然宕机的话，也能尽量保证数据少丢失。为了保证对数据的更新是顺序性的，这里把数据入库的操作使用队列单线程化。逻辑服务器与数据库服务器通过Java的TCP/IPSocket进行长连接，而且为了防止由于意外原因导致连接中断，在逻辑服务器与数据库服务器之间加入了一个心跳连接，这样短暂的中断可以被很快恢复，防止数据的丢失。

**日志服务器**

处理玩家日志的入库。日志入库方便游戏运营管理游戏，统计玩家信息。当玩家人数比较多的时候，日志也会占用很多资源。所以把日志从逻辑服务器也分开了，因为日志只是插入操作，所以可以开几个线程进行并发插入到数据库。线程数要根据你数据库的连接池的最大连接数进行设置，要不然会导致连接资源被占完，数据插入不了数据库。

**注意事项**

在游戏服务器开发中，有几个需要注意的问题。

**通信协议**

开始的时候，我们为了快速开发，采用了JSON的变长协议处理方式，即把要传送的数据编码成json的字符串，再把json字符串转化为字节数据，传输过程中包的总结构为：总包长度（int四个字节）+消息长度（int四个字节）+消息体，即数据长度，n个字节。这样做的好处是可以快速开发，缺点是在传输过程中无效的字节太多。而且这部分完全可以用代码自动完成。后来我们采用Java的反射机制，从定义好的xml描述协议文件中读取传输的内容格式，自动化生成传输的对象，在发送信息时，根据这个对象再把数据转化为二进制的数据流，解析的时候，同样也根据xml的描述文件，按顺序读取数据并转化为对象的JavaBean对象。如果时间充足，在游戏开发前期应该把这个做好。

**多线程并发**

游戏服务器是一个多用户的环境，其中多线程是必不可少的，它可以提交程序对CPU的利用率，提高处理性能。但它也有一个致命的缺点，就是在多线程下，数据同步的问题。因为在目前多核CPU下，线程算得上是可以并行执行的了。比如竞技场中的排行榜，每个玩家的名次变化都会对排行榜进行操作。如果不考虑数据同步的话，每个玩家可以随意更新排行榜，那这个排行榜的数据就会非常乱，名次也不正确。这个时间就需要保证在一个玩家更新排行的时候，其他玩家不能更新，只能阻塞等待。一般有两种方法可以解决：1.直接使用锁，当一个玩家更新排行榜时，使用锁锁定排行榜集合，让其他玩家不能再对排行榜操作，Java有自带的两种方式，非常方便，一个是Lock接口，一个是Synchronized；2.使用乐观同步，这种方式需要自己额外实现，之所以说是乐观，是因为它有可能执行失败。原理是当我取数据时，获得一个数据的一个版本号，而当写入数据时，如果版本一致，可写入，如果版本不一致，就需要重新获取数据，执行逻辑，直到版本一致后写入。可以设定重复次数，达到这个次数后，还没有成功就判定失败。根据我们目前的运行环境，我们采用了第一种方式。

**均衡负载**

一台物理服务器的处理能力是有限的，对于可能支持数据众多的游戏服务器来说，分布式部署和动态添加服务器是不可缺少的。在逻辑上，可以把需要集中处理，与逻辑运算关系不大的模块单独部署。比如登录服务器、地图服务器、聊天服务器、数据库服务器等。像我们把登录服务器和数据库服务器分离开就是为了减少逻辑服务器的压力。

**缓存的设计**

起初，为了快速敏捷开发，我们采用了一级缓存方式，即图1中的Redis缓存，它是一个分布式的缓存，内部通过Socket连接。在玩家第一次进入游戏的时候会把玩家数据从数据库加载到Redis缓存之中，再取数据只从缓存中取。后来为了更加提高处理速度，增加了二级缓存，即内存缓存，利用Java提供的Map、List等集合保存数据，开发了一个基于内存的缓存构架MemoryCacheTool，对外只提供操作接口。由于是直接从内存中读取或写入数据，其速度相对于Redis提高大约20%左右。

**总结**

Java是一门安全、高效、跨平台的语言，它在游戏服务器开发过程中可以提高开发效率，减少异常，增强程序的健壮性。它很容易实现各个功能的衔接，方便后期的维护。



王广帅

**作者简介：**王广帅，网名wgslucky，2011年毕业开始参加工作，一直从事于游戏服务器的开发工作。现任北京触云科技有限公司服务器端技术主程，是一名技术狂，热爱读书，喜欢挑战，热衷与技术同道分享交流游戏开发方面的技术。