

# 《网络游戏核心技术与实战》

## 图书基本信息

书名：《网络游戏核心技术与实战》

13位ISBN编号：9787115349355

出版时间：2014-4

作者：中嶋谦互

页数：443

译者：毛姝雯,田剑

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu111.com](http://www.tushu111.com)

# 《网络游戏核心技术与实战》

## 内容概要

本书围绕实时通信、大数据量通信的多人网络游戏开发，详细介绍了普通开发者如何在不使用昂贵的中间件或者特殊开发环境的基础上，独自从零开始实现有趣的多人网络游戏系统，并讲解了C/S MMO游戏和P2P MO游戏这两个典型的开发案例。同时还从游戏运营和基础设施架构等角度，展现了支持网络游戏技术的全貌。

# 《网络游戏核心技术与实战》

## 作者简介

作者简介：

中嶋谦互

从小学开始学习游戏编程，进入大学后正式参与游戏开发。1996年开发了世界上第一款使用Java Applet的MMORPG，1998年在Windows平台发布后续版本Lifestorm系列，大受欢迎。2001年研发了开发在线游戏的中间件VCE，包括自己开发的gumonji在内，有大约50家公司使用该工具。目前已是两个孩子的父亲，在抚养孩子的同时继续从事开发工作。

译者简介：

毛姝雯（译者）

现任职于IBM。对游戏设计和开发有着浓厚兴趣，对UI设计也有一定研究。译著《iOS用户体验设计》，合译《iOS 5基础教程》。

田剑（译者）

资深游戏开发工程师，曾在日本东京OptiMedia株式会社工作4年，游戏开发实战经验丰富。目前就职于D2C China。对于移动互联网行业新趋势、新技术有强烈兴趣。目前专注于iOS、Android和Windows Phone等移动平台的游戏开发。

## 书籍目录

|  |     |
|--|-----|
| 第0章 [快速入门] 网络游戏编程网络和游戏编程的技术基础              | 001 |
| 0.1 网络游戏开发者所需了解的网络编程基础                     | 003 |
| 0.1.1 网络编程是必需的                             | 003 |
| 0.1.2 网络编程与互联网编程                           | 003 |
| 0.1.3 互联网编程的历史和思想                          | 004 |
| 0.1.4 OSI 参考模型——透明地处理标准和硬件的变化              | 004 |
| 0.1.5 网络游戏系统及其层次结构                         | 005 |
| 0.1.6 套接字API 的基础知识                         | 006 |
| 0.1.7 网络游戏和套接字API——使用第4 层的套接字API           | 007 |
| 专栏 网络编程的特性和游戏架构的关系                         | 008 |
| 0.2 套接字编程入门——处理多个并发连接、追求性能                 | 010 |
| 0.2.1 通信链路的确定（复习）                          | 010 |
| 0.2.2 套接字API 基础——一个简单的ECHO 服务器、ECHO 客户端示例  | 011 |
| 0.2.3 TCP 通信链路的状态迁移和套接字API                 | 012 |
| 0.2.4 处理多个并发连接——通向异步套接字API 之路              | 015 |
| 0.2.5 同步调用（阻塞）和线程                          | 016 |
| 0.2.6 单线程、非阻塞、事件驱动——使用select 函数进行轮询        | 017 |
| 0.2.7 网络游戏输入输出的特点——单线程、事件驱动、非阻塞            | 018 |
| 0.2.8 网络游戏和实现语言                            | 018 |
| 0.2.9 充分发挥性能和提高开发效率——从实现语言到底层结构            | 018 |
| 0.2.10 发挥多核服务器的性能                          | 020 |
| 专栏 输入输出的实现方针和未来提高性能的可能性                    | 020 |
| 0.2.11 多核处理器与网络吞吐量——网络游戏与小数据包              | 021 |
| 0.2.12 简化服务器实现——libevent                   | 025 |
| 0.3 RPC 指南——最简单的通信中间件                      | 026 |
| 0.3.1 通信库的必要性                              | 026 |
| 0.3.2 网络游戏中使用的RPC 的整体结构                    | 028 |
| 0.3.3 [补充] UDP 的使用                         | 030 |
| 0.4 游戏编程基础                                 | 031 |
| 0.4.1 游戏编程的历史                              | 031 |
| 0.4.2 采用“只要能画点就能做出游戏”的方针来开发入侵者游戏           | 031 |
| 0.4.3 游戏编程的基本剖析                            | 032 |
| 0.4.4 游戏编程精粹——不使用线程的“任务系统”                 | 040 |
| 0.4.5 两种编程方法的相似性——不使用线程                    | 040 |
| 0.5 小结                                     | 041 |
| 专栏 确保开发效率和各平台之间的可移植性                       | 041 |
| 第1章 网络游戏的历史和演化游戏进入了网络世界                    | 043 |
| 1.1 网络游戏的技术历史                              | 045 |
| 1.1.1 网络游戏出现前的50 年                         | 045 |
| 1.1.2 20 世纪50 年代前：计算机诞生                    | 045 |
| 1.1.3 20 世纪50 年代：早期的电子游戏                   | 046 |
| 1.1.4 20 世纪60 年代：各种颇具影响的机器登上历史舞台           | 046 |
| 1.1.5 20 世纪70 年代：网络游戏的基本要素                 | 050 |
| 1.1.6 20 世纪80 年代：网络对战游戏登场                  | 051 |
| 1.1.7 20 世纪90 年代：游戏市场扩大                    | 052 |
| 1.1.8 本世纪前10 年的前期：网络游戏商业化                  | 055 |
| 1.1.9 本世纪前10 年的后半期：基于Web 浏览器的MMOG 在商业上获得成功 | 056 |

|        |                                  |                        |
|--------|----------------------------------|------------------------|
| 1.1.10 | 2010 年之后：究竟会出现什么样的游戏呢？           | 057                    |
| 1.2    | 从技术变迁看游戏文化和经济圈                   | 057                    |
| 1.2.1  | 解读技术发展图                          | 057                    |
| 1.2.2  | 3 个圈（三大范畴）                       | 057                    |
| 1.2.3  | 两个游戏经济 / 文化圈                     | 059                    |
| 1.2.4  | 文化、经济与技术的关系                      | 061                    |
| 1.3    | 小结                               | 062                    |
| 专栏     | 成为出色的网络游戏开发程序员的条件                | 062                    |
| 第2章    | 何为网络游戏网络游戏面面观                    | 063                    |
| 2.1    | 网络游戏术语的定义                        | 065                    |
|        | 网络游戏的4 个层面                       | 065                    |
| 2.2    | 网络游戏的物理层面                        | 066                    |
| 2.2.1  | 物理构成要素                           | 066                    |
| 2.2.2  | 物理模式 / 物理上的网络构成                  | 067                    |
| 2.3    | 网络游戏的概念层面                        | 069                    |
| 2.3.1  | 网络游戏及其基本结构                       | 069                    |
| 2.3.2  | 游戏进行空间——进行游戏时所需的所有信息             | 070                    |
| 2.3.3  | 游戏的进展——游戏进行空间的变化                 | 070                    |
| 2.3.4  | 共享相同的游戏进展                        | 071                    |
| 2.4    | 网络游戏的商业层面                        | 072                    |
| 2.4.1  | 商业层面的要求                          | 072                    |
| 2.4.2  | 有效地招募测试玩家——网络游戏与测试               | 073                    |
| 2.4.3  | 不断更新——网络游戏的运营和更新                 | 074                    |
| 2.4.4  | 节约服务器数量和带宽——网络游戏开支的特殊性           | 076                    |
| 2.4.5  | 从小规模开始，确保可扩展性——将风险降到最低，不要错过取胜的机会 | 077                    |
| 2.4.6  | 提供多种收费方式——收费结算方式的变化              | 077                    |
| 2.4.7  | 低价、快速地根除攻击者——攻击、非法行为及其对策         | 079                    |
| 2.4.8  | 减少服务器停止的次数和时间——不要让玩家失望           | 080                    |
| 2.4.9  | 反馈游戏结果——日志分析和结果的可视化              | 082                    |
| 2.4.10 | 更容易地与其他玩家相遇——玩家匹配                | 084                    |
| 2.5    | 网络游戏的人员和组织                       | 085                    |
| 2.5.1  | 与网络游戏服务的运营相关的人员                  | 085                    |
| 2.5.2  | 网络游戏服务运营的3 项专门职责                 | 086                    |
| 2.5.3  | 开发团队                             | 087                    |
| 2.5.4  | 运维团队                             | 089                    |
| 2.6    | 网络游戏程序员所需的知识                     | 090                    |
| 2.6.1  | 网络游戏程序员所需的技术和经验                  | 090                    |
| 2.6.2  | 各种网络游戏开发知识                       | 095                    |
| 2.7    | 支持网络游戏的技术的大类                     | 095支持网络游戏的技术的4 种形式 096 |
| 2.8    | 影响开发成本的技术要素                      | 097                    |
| 2.8.1  | 网络游戏与如今的开发技术                     | 097                    |
| 2.8.2  | 支持网络游戏主体的3 大核心                   | 098                    |
| 2.9    | 小结                               | 102                    |
| 专栏     | 网络游戏编程的最大难点                      | 103                    |
| 第3章    | 网络游戏的架构挑战游戏的可玩性和技术限制             | 109                    |
| 3.1    | 游戏编程的特性——保持快速响应                  | 111                    |
| 3.1.1  | 响应速度的重要性——时间总是不够的                | 111                    |
| 3.1.2  | 将数据存放在内存中的理由——游戏编程真的有三大痛苦吗       | 111                    |
| 3.1.3  | A 每16 毫秒变化一次——处理的信息及其大小          | 111                    |

|        |                                    |     |
|--------|------------------------------------|-----|
| 3.1.4  | B 大量对象的显示——CPU 的处理能力               | 114 |
| 3.1.5  | C 无法预测玩家的操作——游戏状态千变万化              | 117 |
| 3.1.6  | 必须将游戏数据放在CPU 所在的机器上                | 117 |
| 3.2    | 网络游戏特有的要素                          | 118 |
| 3.2.1  | 通信延迟——延迟对游戏内容的限制                   | 118 |
| 3.2.2  | 带宽——传输量的标准                         | 120 |
| 3.2.3  | 服务器——成本、服务器数量的估算                   | 121 |
| 3.2.4  | 安全性——网络游戏的弱点                       | 121 |
| 3.2.5  | 辅助系统（相关系统）                         | 125 |
| 3.3    | 物理架构详解——C/S 架构、P2P 架构              | 125 |
| 3.3.1  | 基本的网络拓扑结构                          | 125 |
| 3.3.2  | 物理架构的种类                            | 127 |
| 3.3.3  | C/S 架构——纯服务器型、反射型                  | 127 |
| 3.3.4  | P2P 架构                             | 129 |
| 3.3.5  | C/S + P2P 混合型架构                    | 130 |
| 3.3.6  | ad-hoc 模式                          | 131 |
| 专栏     | 游戏客户端是什么                           | 131 |
| 3.4    | 逻辑架构详解——MO 架构                      | 132 |
| 3.4.1  | MO、MMO 是什么？——同时在线数的区别              | 132 |
| 3.4.2  | MO 架构、MOG                          | 133 |
| 3.4.3  | 同步方式——获得全体玩家的信息后，游戏才能继续            | 133 |
| 3.4.4  | 同步方式/ 全网状结构的实现——所有终端都拥有主数据         | 133 |
| 3.4.5  | 同步方式/ 星型结构——暂时将输入信息集中到服务器上         | 139 |
| 3.4.6  | 异步方式——接受各终端上游戏状态的不一致               | 142 |
| 3.4.7  | 三大基本要素：自己、对手、环境——异步实现的指导方针         | 143 |
| 3.4.8  | A 自己和对手——对战游戏和玩家之间往来数据的抽象程度        | 144 |
| 3.4.9  | 保持结果一致性的方法——两种覆盖方式                 | 148 |
| 3.4.10 | B 自己和环境——可使用物品的格斗游戏和互斥控制           | 151 |
| 3.4.11 | 互斥控制的实现——采用与同步方式类似的机制来实现异步方式       | 152 |
| 3.4.12 | 状态会自动变化的环境——静态环境和动态环境              | 158 |
| 3.4.13 | C 对手和环境的关系                         | 162 |
| 3.5    | 逻辑架构详解——MMO 架构                     | 163 |
| 3.5.1  | MMO 架构、MMOG——在大量玩家之间共享长期存在的游戏过程    | 164 |
| 3.5.2  | MMOG 的结构                           | 165 |
| 3.5.3  | 大型多人网络游戏（MMO）                      | 168 |
| 3.6    | 小结                                 | 168 |
| 专栏     | 设法改善网页游戏的画面显示间隔                    | 169 |
| 第4章    | [ 实践] C/S MMO 游戏开发长期运行的游戏服务器       | 171 |
| 4.1    | 网络游戏开发的基本流程                        | 173 |
| 4.1.1  | 项目文档/ 交付物                          | 173 |
| 4.1.2  | 开发的进行和文档准备的流程                      | 175 |
| 4.1.3  | 技术人员的文档/ 交付物                       | 176 |
| 4.2    | C/S MMO 游戏的发展趋势和对策                 | 177 |
| 4.2.1  | C/S MMO 游戏的特点                      | 177 |
| 4.2.2  | C/S MMO 架构（MMO 架构）特有的游戏内容          | 177 |
| 4.3    | 策划文档和5 种设计文档——从虚构游戏K Online 的开发中学习 | 179 |
| 4.3.1  | 考虑示例游戏的题材                          | 179 |
| 4.3.2  | 详细设计文档                             | 180 |
| 4.3.3  | MMOG 庞大的游戏设定                       | 181 |

|        |  |     |
|--------|--|-----|
| 4.3.4  | 5 种设计文档                                    | 182 |
| 4.3.5  | 设计上的重要判断                                   | 182 |
| 4.4    | 系统基本结构图的制定                                 | 184 |
| 4.4.1  | 系统基本结构图的基础                                 | 184 |
| 4.4.2  | 服务器必须具有可扩展性——商业模式的确认                       | 184 |
| 4.4.3  | 各种瓶颈——扩展方式的选择                              | 184 |
| 专栏     | MMO 客户端特有的渲染性能瓶颈                           | 186 |
| 4.4.4  | 解决游戏服务器/数据库的瓶颈                             | 187 |
| 4.4.5  | 什么都不做的情况(1 台服务器负责整个游戏世界)                   | 188 |
| 4.4.6  | 空间分割法——解决游戏服务器的瓶颈                          | 189 |
| 4.4.7  | 实例法——解决游戏服务器的瓶颈                            | 190 |
| 4.4.8  | 平行世界方式——解决数据库瓶颈                            | 191 |
| 4.4.9  | 同时采用多种方法——应对越来越多的玩家                        | 193 |
| 4.4.10 | 各种方式的引入难度                                  | 194 |
| 4.4.11 | 各个世界中数据库(游戏数据库)服务器的绝对性能的提高                 | 194 |
| 4.4.12 | K Online 的设计估算——首先从同时在线数开始                 | 195 |
| 4.4.13 | 根据游戏逻辑的处理成本来估算——敌人的行动算法需要消耗多少CPU           | 196 |
| 4.1.14 | 根据游戏数据库的处理负荷进行估算——找到“角色数据的保存频率”与“数据库负荷”的关系 | 198 |
| 4.4.15 | 可扩展性的最低讨论结果,追求进一步的用户体验                     | 199 |
| 4.4.16 | 服务器的基本结构,制定系统基本结构图                         | 200 |
| 4.5    | 进程关系图的制定                                   | 201 |
| 4.5.1  | 进程关系图的准备                                   | 201 |
| 4.5.2  | 服务器连接的结构——只用空间分割法                          | 202 |
| 4.5.3  | 服务器连接的结构——使用平行世界方式和空间分割法                   | 203 |
| 4.5.4  | 使用平行世界方式进行扩展的关键点                           | 204 |
| 4.6    | 带宽/服务器资源估算文档的制定                            | 205 |
| 4.6.1  | 以进程列表为基础估算服务器资源                            | 205 |
| 4.6.2  | 以CPU为中心的服务器和以存储为中心的服务器                     | 208 |
| 4.6.3  | 服务器资源的成本估算——首先从初期费用开始                      | 208 |
| 4.6.4  | 带宽成本的估算                                    | 209 |
| 4.6.5  | 带宽减半的方针——首先是调整策划,然后在程序上下功夫                 | 210 |
| 4.6.6  | 策划内容的分析对带宽的降低很有效                           | 211 |
| 4.7    | 协议定义文档的制定——协议的基本性质                         | 211 |
| 4.7.1  | 协议定义文档基础                                   | 212 |
| 4.7.2  | “协议的基本性质”的要点                               | 212 |
| 4.7.3  | 协议的种类、以及进程之间关系的种类                          | 213 |
| 4.7.4  | 8 种类型的协议                                   | 213 |
| 4.7.5  | C/S MMO 采用TCP                              | 214 |
| 4.7.6  | 与“协议的基本性质”的对应                              | 214 |
| 4.8    | 协议定义文档——协议的API规范(概要)                       | 217 |
| 4.8.1  | 协议的实现原则                                    | 217 |
| 4.8.2  | 8 种协议的功能/形式概述                              | 221 |
| 4.9    | 协议定义文档——协议的API规范(详细)                       | 223 |
| 4.9.1  | 协议API规范(详细)的制定                             | 224 |
| 4.9.2  | API的函数定义                                   | 224 |
| 4.9.3  | 常量定义                                       | 229 |
| 4.9.4  | API的调用时序                                   | 230 |
| 4.9.5  | 时序图制定的要点                                   | 238 |

|         |  |     |
|---------|--|-----|
| 4.10    | 协议定义文档——数据包的格式                             | 238 |
| 4.10.1  | C/S MMO 主要采用TCP (复习)                       | 238 |
| 4.10.2  | C/S MMO 使用包含专用字节数组的二进制协议                   | 239 |
| 4.10.3  | 二进制协议的实现——首先从术语的整理开始                       | 239 |
|         | 专栏 C/S MMO 的压缩和加密                          | 242 |
| 4.11    | 数据库设计图                                     | 242 |
| 4.11.1  | 要在编程之前进行对重要的表进行设计                          | 242 |
| 4.11.2  | C/S MMO 中的数据库实现的历史变迁                       | 243 |
| 4.11.3  | 整理K Online 所需的表                            | 245 |
|         | 专栏 百花缭乱的KVS                                | 246 |
| 4.11.4  | 数据库性能预测                                    | 250 |
| 4.12    | 服务器/ 客户端软件 + 中间件——实践中不可或缺的开发基础             | 254 |
| 4.12.1  | 网络游戏的中间件                                   | 254 |
| 4.12.2  | 开发的基础软件——可以立刻尝试的C/S MMO 开发体验               | 256 |
| 4.13    | 程序开发中的基本原则                                 | 258 |
| 4.13.1  | 如何开始编程、如何继续编程                              | 258 |
| 4.13.2  | 数据结构优先原则——基本原则                             | 259 |
| 4.13.3  | 实现数据结构之前的讨论——出现在画面上和不出现在画面上的元素             | 262 |
| 4.13.4  | 维持可玩状态的原则——基本原则                            | 263 |
| 4.13.5  | 后端服务器的延后原则——基本原则                           | 264 |
| 4.13.6  | 持续测定的原则——基本原则                              | 264 |
| 4.14    | C/S MMO 游戏K Online 的实现——编程开始!              | 266 |
| 4.14.1  | 开发的安排                                      | 266 |
| 4.14.2  | K Online 中的分工计划                            | 267 |
| 4.14.3  | K Online 中“ 框架阶段 ”和“ 原型阶段 ”的区别             | 268 |
| 4.14.4  | [ 步骤1 ~ 2] 框架 ~ 原型阶段                       | 269 |
|         | 专栏 每一步的进度管理形式                              | 272 |
| 4.14.5  | “ 不实际运行起来是不会理解的! ”——游戏开发的特殊性               | 274 |
|         | 专栏 C/C++ 以外的语言                             | 275 |
| 4.14.6  | 框架的整体结构                                    | 276 |
|         | 专栏 VCE 是什么                                 | 278 |
| 4.14.7  | 以怎样的顺序来编写代码                                | 280 |
| 4.14.8  | 首先编写协议定义文件k.xml——开发流程                      | 280 |
| 4.14.9  | 协议定义的要点                                    | 281 |
| 4.14.10 | 通信连通确认: ping 函数                            | 281 |
| 4.14.11 | 账户登录和账户认证: signup 函数、authentication 函数     | 283 |
| 4.14.12 | 角色创建: createCharacter 函数                   | 284 |
| 4.14.13 | 登录: login 函数                               | 285 |
| 4.14.14 | 在地面上移动: move 函数、moveNotify                 | 286 |
| 4.14.15 | 编写gmsv/Makefile——开发流程                      | 293 |
| 4.14.16 | 从示例中复制gmsv/climain.cpp 和gmsvmain.cpp——开发流程 | 293 |
|         | 专栏 dbsv 服务器代码的自动生成                         | 297 |
| 4.14.17 | 自动测试客户端autocli 的实现——开发流程                   | 300 |
|         | 专栏 gmsv 中线程的使用                             | 305 |
| 4.14.18 | 图形客户端cli 的创建和运行确认——开发流程                    | 307 |
| 4.14.19 | 框架之后的开发——开发流程、后续事项                         | 310 |
| 4.15    | 总结   | 310 |
| 第5章     | [ 实践] P2P MO 游戏开发                          |     |
|         | 没有专用服务器的动作类游戏的实现                           | 311 |



|       |  |     |  |
|-------|--|-----|--|
| 5.1   | P2P MO 游戏的特点和开发策略                      | 313 |  |
| 5.1.1 | P2P MO 和动作类游戏——游戏的状态频繁发生改变             | 313 |  |
| 5.1.2 | RPC 和共享内存                              | 313 |  |
| 5.1.3 | P2P MO 游戏的特点——和c/s MMO 游戏的比较和难点        | 314 |  |
| 5.1.4 | P2P MO 游戏的优点                           | 316 |  |
| 5.1.5 | 从概要设计开始考虑 [ 多人游戏模式 ]                   | 316 |  |
| 5.2   | J Multiplayer 游戏开发案例的学习——和K Online 的不同 | 316 |  |
| 5.2.1 | J Multiplayer ——和K Online 的比较          | 317 |  |
| 5.2.2 | P2P MO 游戏开发的基本流程                       | 317 |  |
| 5.2.3 | P2P MO 游戏开发的交付产品——开发各个阶段需要提交的资料        | 317 |  |
| 5.2.4 | 和C/S MMO 的数据量 / 规模的比较                  | 319 |  |
| 5.3   | P2P MO 游戏的设计资料                         | 319 |  |
| 5.3.1 | 系统基本结构图                                | 320 |  |
| 5.3.2 | 进程关系图                                  | 320 |  |
| 5.3.3 | 带宽 / 服务器资源计算资料                         | 323 |  |
| 5.3.4 | 通信协议定义资料和API 规格                        | 323 |  |
| 专栏    | 什么是“ 游戏逻辑 ”                            | 324 |  |
| 5.3.5 | 带宽消耗量的估算                               | 327 |  |
| 5.3.6 | 其他资料                                   | 331 |  |
| 5.4   | 客户端 / 服务器软件 + 中间件、基本原则                 | 332 |  |
| 5.4.1 | P2P MO 开发的最终交付产品                       | 332 |  |
| 5.4.2 | P2P MO 中使用的中间件                         | 333 |  |
| 5.4.3 | 编程时应该注意的基本原则——针对P2P MO 游戏              | 333 |  |
| 5.5   | P2P MO 游戏J Multiplayer 的实现——正式开始编程     | 334 |  |
| 5.5.1 | J Multiplayer 的编程计划                    | 334 |  |
| 5.5.2 | 开发流程——K Online 的回顾                     | 334 |  |
| 5.5.3 | J Multiplayer 开发阶段——开发顺序和内容            | 335 |  |
| 5.5.4 | 第1 阶段的要点                               | 335 |  |
| 5.5.5 | 客户端程序的开发案例                             | 336 |  |
| 5.5.6 | “ 共享内存方式 ” 的实现——开始编码                   | 337 |  |
| 5.5.7 | P2P MO 游戏开发中该如何防止发生竞争状态                | 339 |  |
| 5.5.8 | 共享内存开发方式该如何编码——共享内存开发方式和RPC 开发方式的比较    | 343 |  |
| 5.5.9 | SyncValue 类                            | 347 |  |
| 专栏    | 数据中心的地理位置分布                            | 352 |  |
| 5.6   | 支持C/S MO 游戏的技术 [ 补充 ]                  | 353 |  |
| 5.6.1 | C/S MO 和NAT 问题                         | 353 |  |
| 5.6.2 | 什么是NAT 问题                              | 353 |  |
| 5.6.3 | NAT 遍历——解决NAT 问题建立通信路径的技术              | 353 |  |
| 5.6.4 | NAT 问题的实际解决方法                          | 355 |  |
| 5.6.5 | 中继服务器                                  | 356 |  |
| 5.6.6 | 中继服务器的折衷方案                             | 356 |  |
| 5.7   | 总结                                     | 358 |  |
| 第6章   | 网络游戏的辅助系统完善游戏服务的必要机制....               | 359 |  |
| 6.1   | 辅助系统需要的各种功能                            | 361 |  |
| 6.1.1 | 现有服务提供的辅助系统功能                          | 361 |  |
| 6.1.2 | 现有服务的功能一览                              | 363 |  |
| 6.1.3 | 网页游戏开发方法和客户端游戏的开发方法                    | 365 |  |
| 6.2   | 交流 / 通信功能                              | 365 |  |
| 6.2.1 | 玩家匹配 P2P MO                            | 365 |  |

|        |                                  |     |
|--------|----------------------------------|-----|
| 6.2.2  | 游戏大厅 P2P MO                      | 369 |
| 6.2.3  | 中继服务器 P2P MO                     | 370 |
| 6.2.4  | 聊天 P2P MO C/S MMO                | 371 |
| 6.2.5  | 邮件 P2P MO C/S MMO                | 374 |
| 6.2.6  | 好友列表 P2P MO C/S MMO              | 374 |
| 6.2.7  | 玩家状态 P2P MO C/S MMO              | 375 |
| 6.2.8  | 加锁服务器 P2P MO C/S MMO             | 376 |
| 6.2.9  | 黑名单 P2P MO C/S MMO               | 377 |
| 6.2.10 | 语音聊天 P2P MO C/S MMO              | 378 |
| 6.3    | 游戏客户端实现相关的辅助系统                   | 378 |
| 6.3.1  | 玩家成绩管理 P2P MO C/S MMO            | 378 |
| 6.3.2  | 存储功能 P2P MO                      | 379 |
| 6.3.3  | (游戏客户端)更新 P2P MO C/S MMO         | 380 |
| 6.3.4  | 排行榜 P2P MO C/S MMO               | 381 |
| 6.4    | 运营辅助系统                           | 383 |
| 6.5    | 新闻发布 P2P MO C/S MMO              | 383 |
| 6.5.1  | 付费相关的辅助系统                        | 384 |
| 6.5.2  | 付费认证 P2P MO C/S MMO              | 384 |
| 6.5.2  | 虚拟货币管理 P2P MO C/S MMO            | 387 |
| 6.6    | 专栏 C/S MMO 游戏的收入                 | 388 |
| 6.6.1  | 其他辅助功能                           | 390 |
| 6.6.1  | 游戏数据浏览 / 查询工具 P2P MO C/S MMO     | 390 |
| 6.6.2  | 敏感词过滤 P2P MO C/S MMO             | 392 |
| 6.7    | 本章小结                             | 392 |
| 第7章    | 支持网络游戏运营的基础设施架构、负荷测试和运营          | 393 |
| 7.1    | 基础设施架构的基础知识                      | 395 |
| 7.1.1  | C/S MMO 和P2P MO 的基础设施 (概要)       | 395 |
| 7.1.2  | 基础设施架构需要进行的工作——从开发整体来看           | 395 |
| 7.1.3  | 基础设施的成本估算                        | 396 |
| 7.1.4  | 成本的概念、单位                         | 397 |
| 7.1.5  | 网络游戏服务器在一定程度上可以接收的条件             | 398 |
| 7.1.6  | 硬件、信息设备                          | 398 |
| 7.1.7  | 软件                               | 399 |
| 7.1.8  | 数据中心相关的成本                        | 401 |
| 7.1.9  | 服务费 (数据中心以外)                     | 403 |
| 7.1.10 | 网络带宽费                            | 403 |
| 7.1.11 | 电费                               | 404 |
| 7.2    | 开发者需要知道的基础设施架构技巧                 | 404 |
| 7.2.1  | 服务规模的扩大/缩小——用户数和需要的基础设施规模        | 404 |
| 7.2.2  | 典型的环境                            | 404 |
| 7.2.3  | 负荷曲线                             | 405 |
| 7.2.4  | 面向开发者的基础设施架构要点                   | 407 |
| 7.2.5  | 服务器部署                            | 407 |
| 7.2.6  | 登台环境                             | 409 |
| 7.2.7  | 服务器的监控、生死监控                      | 410 |
| 7.2.8  | 日志输出/管理                          | 411 |
| 7.3    | K Online、J Multiplayer 游戏的基础设施架构 | 412 |
| 7.3.1  | K Online 的基础设施                   | 413 |
| 7.3.2  | J Multiplayer 的基础设施              | 414 |

|       |                             |     |
|-------|-----------------------------|-----|
| 7.4   | 负荷测试                        | 417 |
| 7.4.1 | 负荷测试的准备                     | 417 |
| 7.4.2 | K Online 在生产环境的负荷测试         | 417 |
| 7.4.3 | 负荷测试时使用的服务器监控命令             | 418 |
| 7.4.4 | J Multiplayer 在生产环境下的负荷测试   | 422 |
| 7.5   | 游戏上线                        | 423 |
| 7.5.1 | 游戏上线前——从确认安全设定开始            | 423 |
| 7.5.2 | 游戏上线后——系统监控                 | 424 |
| 7.5.3 | 服务器的组群化                     | 425 |
| 7.5.4 | 故障发生时的应对                    | 427 |
| 7.6   | 本章小结                        | 428 |
| 第8章   | 网络游戏的开发体制团队管理的挑战            | 429 |
| 8.1   | 游戏的策划内容和开发团队 网络游戏特有的挑战      | 431 |
| 8.1.1 | 游戏的策划内容是团队管理的关键             | 431 |
| 8.1.2 | 游戏数据的持久化                    | 431 |
| 8.1.3 | 游戏中玩家之间的关系                  | 432 |
| 8.1.4 | 游戏结果的共享范围                   | 433 |
| 8.1.5 | 聊天系统的内容                     | 434 |
| 8.1.6 | 维护和升级的计划                    | 434 |
| 8.1.7 | 代码规模——如果需要迭代的代码过多就会遇到问题     | 435 |
| 8.2   | 网络游戏开发团队的实际情况——和一般软件开发相同的地方 | 436 |
| 8.2.1 | 工作分配                        | 436 |
| 8.2.2 | 持续提升网络游戏程序员技能的方法            | 438 |
| 8.2.3 | 项目管理术——游戏开发和Scrum           | 440 |
| 8.2.4 | 开发环境的选择                     | 440 |
| 8.2.5 | 项目的移交——理所当然的事情也需要仔细归纳总结     | 441 |
| 8.3   | 本章小结                        | 442 |
| 专栏    | 网络游戏开发的成本                   | 443 |

# 《网络游戏核心技术与实战》

## 精彩短评

- 1、茅塞顿开的一本书。看这边书之前，刚刚摸索着搞定了格斗游戏的对战，虽然感觉上还有不少地方可以提高，但是总是感觉有种隐隐的担心。书刚看过一半的时候，就有种为什么没早看的低落心情。之前纠结，不确定，懵懂的事情，书中都有了比较经验性的说明。虽然不是切实的code，也足以解惑。书中不光有MMO,MO的知识。还有游戏设计，运维等等全部的游戏开发的只是。最为一个2年后台开发，有赚手机游戏开发的coder来说，这本书的含金量超高。
- 2、如果从一个游戏从业者，全面了解一款网游所有细节，这本书讲得还蛮详细的。
- 3、大致带读者过了一遍网游开发的各方面。入门。
- 4、挺不错的
- 5、第三章对MO同步异步方式，第四章对MMO服务器基础文档资源估算，讲解的不错。
- 6、讲述mmo服务器技术最商业化的一本书（看过很多很学术不实用的），没有之一，居然没人读，可惜了。目前很多网游的服务器结构和书中描述一致，作者很严谨。
- 7、仔細的說明网络编程所包含的技術, 適合新手作一個概覽
- 8、比较粗糙，对入门的可能有些很大帮助，适合新手。
- 9、游戏制作的方方面面，不仅仅是技术
- 10、内容很全面，学到很多东西，作者认真严谨的治学精神值得学习
- 11、讲得很粗，大致看了一下
- 12、泛泛而谈，有具体情况的时候又讲得不具有普适性。  
各方面浅尝辄止，对很多关键实现讲得不透，不少地方抛出了问题却没有结论。
- 13、11区人民写出来的东西确实质量有保障
- 14、我读过了。。
- 15、科普书，都不深，不过很全，很多东西自己玩wow的时候也想到了。
- 16、读了三分之一，后面的先放一放了，岛国的严谨态度用在技术书籍上再适合不过了
- 17、不推荐。
- 18、适合刚入行的人看，方方面面都讲到了，但是都是点到为止。想要深入还要看其他的书。

# 《网络游戏核心技术与实战》

## 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:[www.tushu111.com](http://www.tushu111.com)