

# インターネットゲームサーバの実装と試行運用

## Implement and Experiment of the Internet Game Server

伊藤琢巳, 小池晃弘, 佐々木広章

marupe@nenemone.com

### 概要

インターネット上で動作するゲームサーバシステムに関して、その機能について検討し実装を行った。本論文ではインターネットゲームサーバシステムの特長について説明を行う。さらに、本サーバシステム試験のため、モンスターメーカーファンクラブと連携しカードゲーム「モンスターメーカー」のサーバ・クライアントシステムを開発し、一定期間の運用をおこなった。その結果と得られた改善点についても述べる。

### 1. はじめに

様々なサービスがインターネット上で提供されてきた。ネットワークゲームもその一つであり、古くは1992年に始まったIGS(Internet GO Server)などがある。日本国内でもインターネットが普及し始めた1990年代後半からJava将棋[1](1996年)、東風荘[2](1997年)といったネットワーク対戦型のゲームが登場している。

ADSLやCATVさらに光といったインターネット接続環境の充実、そして接続コストの大幅な低下、さらに計算機の性能向上といった要因が重なり、今では家庭用ゲーム機上でもインターネットでのゲームサービスが提供されるようになり、その種類も多岐にわたっている。

インターネット上で提供されるゲームはその特徴によっていくつか分類される。

(a) ボードゲーム・カードゲーム、(b) リアルタイムゲーム、(c) 大規模オンラインロールプレイングゲーム(MMORPG)。

我々が今回対象とするのは(a)のカテゴリである。(b)のリアルタイムゲームでは、NW上の遅延やデータ転送速度といったサーバ以外の問題点が存在し、(c)のMMORPGではサーバの設備自体がかなり大規模となるため試作が困難であることによる。(a)のカテゴリであれば、NWの遅延や転送速度の影響が少なく、小規模なシステムでの動作確認が可能となるからである。

今回我々がインターネットゲームサーバシステム

(以下、サーバシステムと呼ぶ)を構築するにあたり留意したのは以下の3点である。

- ・多くのゲームに少ない変更で対応できること
- ・拡張性を確保すること(複数のサーバに分散配置可能)
- ・オペレーションを容易にすること

本論文では、サーバシステムの特長について説明を行う。さらに、サーバシステムの試験のため、モンスターメーカーファンクラブと連携しカードゲーム「モンスターメーカー」のサーバ、クライアントシステムを開発し、一定期間の運用をおこなった。その結果と得られた改善点についても述べる。

### 2. サーバシステム

サーバシステムはUNIX上(FreeBSD)で構築されている。図2.1にサーバシステムの構成を示す。図中のサーバはUNIXでのプロセスに相当する。一台の計算機上で全てのサーバを配置することも、複数台の計算機に分散して配置することも可能である。

サーバシステムは、

- ・サービス系
  - ・バックヤード系
  - ・ツール類
- の3つに分類される。

サービス系サーバは認証サーバ、ゲームサーバ、管理用WWWサーバの3種類で構成される。認証サー

サーバはクライアントからのサービス接続要求を受け付け、ユーザ ID、パスワードによる認証を行い正しく認証

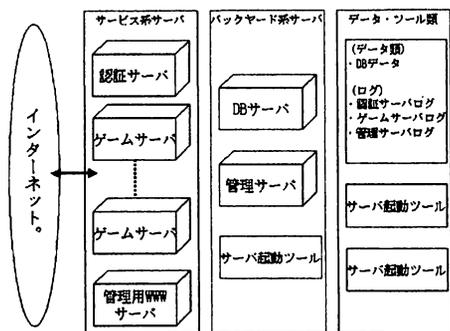


図 2.1 ゲームサーバシステム構成

された場合に現在利用可能なゲームサーバ一覧をクライアントに送信する。ゲームサーバは認証サーバで認証済みのクライアントから接続を受け付け、クライアント間の通信を仲介する。管理サーバは、本システムの運用管理に関する GUI をシステム管理・運用者に提供する。

バックヤード系サーバは、DB サーバ、管理サーバから構成される。DB サーバは、ユーザ情報及び各種サーバの状態を記録する。管理サーバは、認証サーバ及びゲームサーバの起動管理に使われる。

ツール類は、サーバが出力するログの管理ツール、サーバ監視ツールから構成される。

サーバシステムに加えて、クライアントソフトの開発を容易にするため WINDOWS (VC++) 用のライブラリを作成した。ライブラリを利用することでクライアント開発者は、サーバとの通信に気を使うことなくターゲットとなるゲームに通信機能を組み込むことが可能となる。

## 2.1 サーバ・クライアント通信モデル

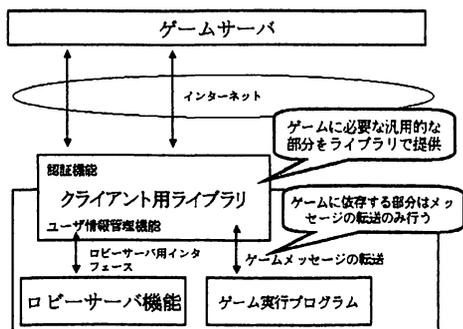


図 2.2 サーバ・クライアント通信モデル

サーバ・クライアントの通信モデルを図 2.2 に示す。サーバ・クライアント間は TCP/IP による通信を行う。

クライアント側からサーバ側に TCP/IP 接続を行うことによってクライアントが NAT を用いてインターネットに接続している場合でもサーバを利用することができる。

クライアントはまず認証サーバに接続し、ユーザ ID とパスワードによる認証を行う。認証が成功した後認証サーバから接続可能なゲームサーバの一覧とワントタイムパスワードが送信され、認証サーバとの通信が完了する。ワントタイムパスワードはユーザの多重ログインを防ぐために用いられる。

ユーザはクライアントが認証サーバから受け取ったゲームサーバリストの中からゲームサーバをひとつ選択してゲームサーバに接続する。ゲームサーバとクライアントの通信は、ユーザ情報管理、マッチメイク (ゲームを開始するまでの対戦者募集とゲームの開始)、ユーザ間の会話、さらにゲーム情報送信の 4 種類に分類される。ゲーム情報送信において、ゲームサーバはあるクライアントから送信されてきたゲーム情報を他のクライアントに送信するだけでその内容には一切関知しない。このような方式をとることによって、ゲームサーバはクライアント上で動作するゲームの種類に依存することなしに、ユーザのマッチメイクや会話といったゲームに共通で必要となる機能をサポートする。

## 2.2 サーバシステムの管理と監視

サーバシステムの管理は http を用いて行う。サーバ管理者は www ブラウザを介して、認証サーバ/ゲームサーバの起動・停止、ユーザ情報登録・変更・削除、サーバログ管理を行うことができる。サーバシステムの運用・管理を 24 時間行うことを想定すると、わかりにくい UNIX コマンドを管理者が直接扱うことは好ましくない。管理者がシステムに関する深い知識を持っていないことも考えられるし、入力ミスによってシステム全体に悪影響を及ぼす可能性も否定できない。

認証サーバ・ゲームサーバを起動する計算機では管理サーバが起動している必要がある。管理サーバが起動する際、その存在(サービスする IP アドレス)を DB サーバに登録する。登録された管理サーバ情報を参照することで、管理ツールからゲームサーバを立ち上げる場合、どの計算機上で立ち上げるかを選択することが可能となり、指定した計算機上の管理サーバにゲームサーバの立ち上げを依頼し、ゲームサーバが起動される。

認証サーバやゲームサーバに関する情報 (IP アドレス、サービスポート、その他) も DB サーバ上で管

理される。

複数の計算機で管理サーバを立ち上げることによって、ゲームサーバを複数の計算機に分散して配置することができる。この方式によって同時参加人数が多くなっても容易にサーバシステムの拡張が可能となる。

サーバシステムの監視は cron によって定期的に行われる。認証サーバ/ゲームサーバは管理ツールから動的に起動されるため、DB から移動しているサーバの IP アドレス/ポートを取得しそれぞれのポートに直接アクセスすることで監視を行う。監視の結果サーバからの反応が無い場合は、管理サーバを介して該当サーバの再起動を実施する。

### 3. サーバシステム試行運用

サーバシステムの試行運用を行うにあたり、「モンスターメーカーファンクラブ」(以下 MMFC) [3]に協力をいただき、カードゲーム「モンスターメーカー」をクライアントとして実装した。「モンスターメーカー」は、1980 年代後半に登場したカードゲームで、洞窟に居る怪物と戦って宝を持ち帰ることを2~6人で競争するゲームである。詳細は[3]を参照されたい。

「モンスターメーカー」を対象ゲームとして選択した理由を述べる。我々がカードゲームとしての「モンスターメーカー」をプレイする過程で、問題と考えた点を以下に示す。

- ・ 「モンスターメーカー」は2人でもプレイできるが、3人以上だと特に面白いゲームであるがために、一緒にゲームをするメンバーを集めるのが容易でないこと
- ・ ルールはA4の紙2枚程度のものであるが初めてゲームをプレイする人に説明する場合にはかなり時間がかかってしまうこと
- ・ ルールの解釈がプレーヤによって異なることがありゲームの進行中にトラブルが発生する可能性があること

インターネットによるゲームサーバ・クライアントシステムを用いることは、これらの問題点を解決するよい方法である。インターネットを介することで、参加者は物理的に同じ場所に集合する必要がなくなった。ルールをよく知らない場合でも基本的なことは全てクライアントが処理を行い、ルールに外れた動作はクライアントが禁止するため、ゲームル

ールを熟読する必要が無くなった。また、クライアントがルールの解釈を行うため、プレーヤ同士でのルール解釈に関するトラブルが生じなくなった。さらに、カードを配る・めくるといった動作がなくなるためゲーム時間の短縮も期待できる。

試行運用では、MMFC 会員約 2000 名の中から希望者にクライアントソフトを配布して自由に遊んでもらう事とした。案内はMMFCのHPにて行い、期間中に登録されたユーザ数は44、合計アクセス回数は1722回であった。

会員数のわりに登録ユーザ数が少ないように見えるが、その理由は2つ考えられる。一つは、2000人の会員のうち定期的にHPにアクセスしてくるいわゆるアクティブな会員は100名前後であること。もう一つは、今回のクライアントソフトウェアが、試験開始時はWINDOWS2K, XPでしか動作しなかったことである。MMFCの掲示板でも話題となったが、WINDOWS98, MEといったOSが現在でも多く使われていることが確認された形となった。

#### 3.1 運用データからの考察

図3.1に日毎のアクセス数、図3.2に24時間別のアクセス数を示す。試験開始直後に非常に多くのアクセスがあったが、その後は数が減少している。時間別の分布では、19時から24時の間のログイン数が多いこと、早朝(3時~6時)のアクセスが少なくなっている。このパターンはゲームに限らず、WWWアクセスなどでもよく見られるパターンと言える。

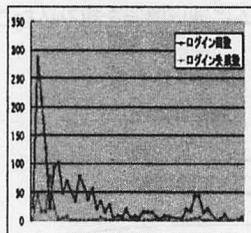


図3.1 ログイン回数の推移

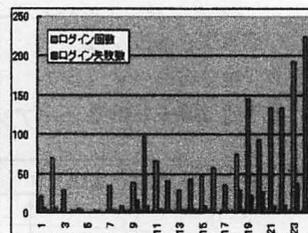
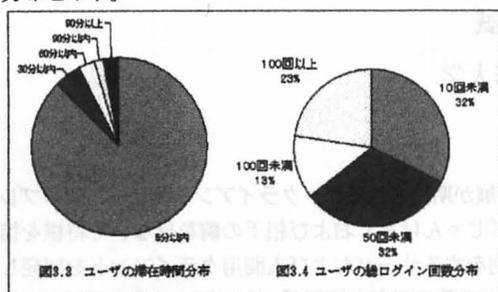


図3.2 時間あたりのログイン数

ログインの失敗はアクセス数の12%を占める。大文字小文字の違いや勘違いによるログイン失敗は10件確認されている。ユーザID・パスワードはcgiを介してユーザに知られるが、それでもユーザIDを間違えるユーザが少なくないことを示している。ゲームクライアントはID、パスワードを記憶しており、2回目以降は手で入力する必要がない。よって認証が失敗する可能性としては2重ログインも考えられる。

図 3.3 にはサーバにログインしたユーザの滞在時間の分布、図 3.4 にはユーザ毎の総ログイン数の分布を示す。



一度ログインしたユーザはその90%が5分以内にログアウトしている。これは、総アクセス数が少ないため、あるユーザがログインした時に他のユーザがログインしていない可能性が高いこと、さらに他のユーザが来るまで待つことのできる時間はせいぜい5分未満であることを示している。実際、今回の場合でも MMFC の掲示板を利用してユーザ同士が待ち合わせるケースがみられた。

ユーザ毎のログイン頻度だが、32%のユーザが10回未満である一方、36%のユーザが50回以上のログインを行っている。ユーザ登録はしてみたものの、サーバでの待ち合わせがうまくいかずにあきらめてしまうユーザも多い一方、ゲームを何回も楽しむユーザも少なくないといえる。

### 3.2 ゲームの評価

「モンスターメーカー」は2人対戦するというよりはむしろ、3人以上でお互いに足を引っ張りあうところがゲームの特徴である。今回3人以上で行われたゲームが全体の75%を占めており、参加者がそれほど多くなかった中でも3人以上での対戦をユーザが望んでいたことを示している。

また、ゲーム時間だが70%のゲームが20分以内で終了していた。カードゲームとしてのプレイ時間は計測していないが、サーバシステムを用いることで明らかにゲーム時間の短縮が図られている。

### 3.3 サーバの評価

参加数が予想より少なかったためサーバの負荷試験になるほどの負荷がかかることはなかった。試験期間中、ソフトバグによる問題が2回発生したが、その後は安定して動作していた。具体的にはDBサーバであるMySQLとのコネクションが切れてしまうこ

とによる認証の不具合が発生した。開発環境ではこの不具合が再現せず、開発マシンと実験マシンの環境の違いによるものとも推測される。対策として定期的にDBサーバへ接続確認を行うことで問題を回避した。

## 4. まとめ

今回ユーザから上がった要望にはサーバ内でのチャットに関するものが多かった。ひとつは、チャットそのものの方式である。WWWブラウザをUIとしてcgiで実装されるチャットに慣れているユーザが多く、ユーザがログインする前のチャットのログを参照したいという要望が多かった。サーバ側でチャットのログを保管しクライアント接続時にログを転送することを検討している。また、異なったゲームサーバに接続しているユーザ間の通信や、知り合いリストの作成なども今後実装していく予定である。

もうひとつ問題となったのが2重ログインである。例えばADSLやダイヤルアップによるインターネット接続を行っている場合、一定時間無通信であると自動でインターネット接続が切断されることがある。しかし、クライアントソフトを停止したわけではないのでサーバからはクライアントが接続されているように見えている。しばらくして同じユーザが再接続を試みようとした場合に2重ログインが発生することになる。この問題を解決するために、定期的にクライアントへの通信を試み、応答が無い場合は切断するという方法を検討している。

今回の目的であるゲームサーバの機能からは外れるものの、計算機によるゲームプレーヤの実装もかなり要望があった。参加人数が少ない場合や、一人で試したい場合の相手として計算機プレーヤを実装することは重要である。

試行運用で開発されていない重要な機能にユーザからの課金があげられる。クレジットカード決済やWebMoneyのようなプリペイド方式などが課金の手段として考えられる。

## 参考文献

- [1] <http://www.javashogi.com/index-j.html>
- [2] <http://mj.giganet.net/>
- [3] <http://monstermaker.jp/>