

# 仮想環境におけるオブジェクトの 密集状態に対応した負荷分散手法\*

遠藤 伶<sup>†</sup> 高木 健士<sup>‡</sup> 北望<sup>‡</sup> 重野 寛<sup>‡</sup>

慶應義塾大学理工学部<sup>†</sup> 慶應義塾大学大学院理工学研究科<sup>‡</sup>

## 1 はじめに

本稿では巨大な環境規模と商業的スケールを持つ Massively Multiplayer Online (MMO) システムに焦点を合わせる。MMO では、プレイヤーは仮想環境上で様々な活動を行い、相互にコミュニケーションを取る事ができる。代表的な MMO は数十万規模のユーザーを常時サポートしている。運営会社が MMO システムに望むのは、低コストで多数のプレイヤーを収容できる事である。そのために Hybrid-P2P を利用した負荷分散法が多く研究されている。しかし、それら既存研究では、仮想空間上で人口が密集する等のオブジェクトの密集状態が考慮されていない。そこで本稿では、仮想環境におけるオブジェクトの密集状態に対応した負荷分散手法を提案する。

## 2 既存研究

MMO システムの最低条件は、プレイヤーが仮想空間上で位置が近いプレイヤーの update メッセージを受け取れる事である。update メッセージとは仮想空間上の位置変更時などに発生するメッセージである。商業化されている MMO はサーバ/クライアント型だが、プレイヤー数に比例してサーバ数を増やす必要があり、コストが非常に高い。そこで Hybrid-P2P アーキテクチャが多く研究されている。仮想環境を分割しピアが分散管理する事で、少ないサーバ数で大規模な仮想環境を維持できる。

Peer Clustering[1] では、仮想環境をピアが扱えるサイズの領域に分割し 128bit の識別子を割り当てる。ピアにも 128bit の識別子が割り当てられ、領域の識別子に最も近い識別子を持つピアが、その領域の管理者となる。管理者となったピアを領域マネージャと呼ぶ。領域マネージャは管理領域の全プレイヤーからメッセージを受け取り、各プレイヤーへ近接者のメッセージを送信している。

一部の領域にプレイヤーが密集すると領域マネージャの処理能力を超える可能性がある。図 1 のように領域マネージャはサーバに領域管理を委託して、システムを維持する。しかしこの手法では、プレイヤーの密集時には負荷の

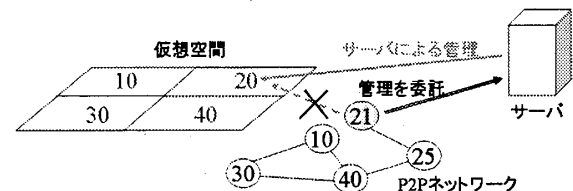


図 1: 領域マネージャのからサーバへの管理移譲

大半がサーバに集中してしまう。また実際の MMO 仮想環境では、パーティという集団単位での行動が多いためプレイヤーが密集する事が多い。そこで、プレイヤーの密集時にも負荷が分散されるようなシステムが必要である。

## 3 パーティLeader による負荷分散手法の提案

上記の問題点を踏まえ MMO 仮想環境のための負荷分散手法を提案する。ランダムに選ばれたピアが領域マネージャとなって分割領域を管理し、管理しきれない場合はサーバに管理を委託する。そして本提案では、パーティが組まれた時にパーティLeader として代表者を決める。パーティLeader がパーティ参加者を一括管理する事で、密集状態での領域マネージャの負荷の分散を行う。

### 3.1 MMO におけるパーティ

MMO には、プレイヤーが任意で少人数の集団を自由に組む事ができる、パーティというシステムがある。パーティを組む事で、パーティ内だけに伝わるパーティチャットを使用できたり、他のメンバの状態を詳しく知る事ができる。MMO で行動する場合は、パーティが組まれる事が多い。そしてパーティ参加者は同じ目的を持って行動しているため仮想空間上で互いに近接している。そのため人口密度に偏りが生じて、いくつかの領域で密集状態が発生する。そこでパーティに代表者を作り、パーティ全体を管理させる事で密集状態に対応する。

### 3.2 Leader による領域マネージャの負荷低減

各プレイヤーは、パーティの代表者なら Leader、代表者以外のパーティ参加者なら Member、そしてパーティを組んでいないなら Normal の状態を取る。図 2 にそれぞれが一人ずつ存在する場合の update メッセージの経路を示す。プレイヤーの外側の円は仮想空間における近接領域であり、近接領域の内側にいるプレイヤーの update メッセージを確実に受け取れなければならない。本提案では

\*Load Distribution Technique Corresponding to Object Density in Virtual Environment

<sup>†</sup>Rei Endo

<sup>‡</sup>Faculty of Science and Technology, Keio University

<sup>‡</sup>Kenji Takagi, Nozomu Kita, Hiroshi Shigeno

<sup>‡</sup>Graduate School of Science and Technology, Keio University

領域マネージャだけでなく Leader プレーヤも全 Normal プレーヤのメッセージを受け取る。図ではプレーヤ N とプレーヤ L は互いに近接していない。しかしプレーヤ M にプレーヤ N のメッセージを転送する必要があるため領域マネージャからプレーヤ N のメッセージを受け取っている。こうする事で、Member プレーヤと領域マネージャの間のメッセージのやり取りを完全に無くせるので、領域マネージャの負荷を減らす事ができる。

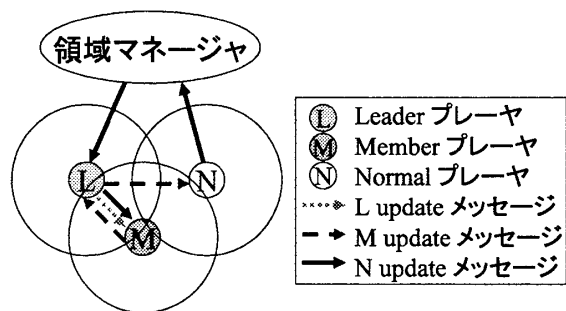


図 2: update メッセージの経路

### 3.2.1 パーティが無い場合の動作

初期状態ではパーティが存在せず、全てのプレーヤが Normal の状態を取る。その場合は領域マネージャが領域内の全てのプレーヤからメッセージを受け取り、そして、全てのプレーヤに近接者のメッセージを転送する。

### 3.2.2 パーティ作成時の動作

本提案ではパーティに勧誘した側が Leader となる。Leader プレーヤは領域マネージャにパーティ成立の通知をする。通知を受け取った領域マネージャは Leader プレーヤに領域内のプレーヤの IP アドレスを渡す。

### 3.2.3 パーティがある場合の動作

領域マネージャは全 Normal プレーヤからメッセージを受け取り、全 Normal プレーヤに対して近接している Normal プレーヤのメッセージを転送する。そして、Leader プレーヤには全 Normal プレーヤのメッセージを転送する。

Leader プレーヤは、領域マネージャから全 Normal プレーヤのメッセージ、他の各 Leader からはそれぞれのパーティ参加者のメッセージ、そして自分の管理パーティの Member プレーヤからメッセージを受け取る。この時点で Leader プレーヤは領域内の全プレーヤのメッセージを保持している事になる。そして、他 Leader プレーヤには自分を含めたパーティ参加者のメッセージ、自分のパーティの Member プレーヤには近接しているプレーヤのメッセージ、Normal プレーヤには近接している自分を含めたパーティ参加者のメッセージを転送する。

以上で、領域内の全てのプレーヤが近接しているプレーヤのメッセージを間違いなく受け取る事ができる。

## 4 評価

### 4.1 シミュレーション条件

管理マネージャとなったピアの処理能力を変化させ、処理能力を超えたためにサーバへ移管する人口を既存手

表 1: シミュレーション条件

分割した領域数	20 × 20 個
総プレーヤ数	2000 人
仮想環境の横断にかかる時間	10 分
ピアの処理能力	250 メッセージ
最大パーティ人数	1~5 人

法、提案手法で測定した。使用したパラメータを表 1 に示す。また試行は 2 時間を 5 回行い、その平均値を算出した。プレーヤは等確率で 1 人から最大パーティ人数までのパーティを組み、パーティを組んだプレーヤは一定の範囲内にまとまって行動するものとする。

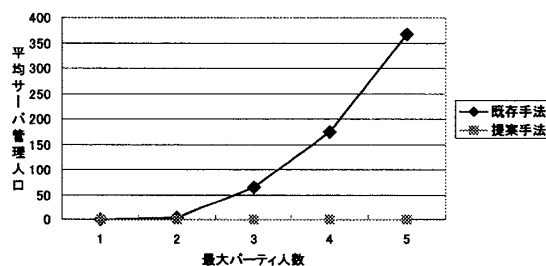


図 3: 最大パーティ人数と平均サーバ管理人口の関係

### 4.2 平均サーバ管理人口

最大パーティ人数と平均サーバ管理人口の関係を図 3 に示す。既存手法では最大パーティ人数に比例してサーバにかかる負荷が増加している。一方、提案手法では最大パーティ人数が増えても平均サーバ管理人口は 0 のままである。よって本提案手法は、パーティが多数組まれた密集状態でも負荷が分散されているので、領域マネージャの負荷を低減し、サーバへの負荷が非常に低く抑えられている事が分かる。

## 5 おわりに

MMO 仮想環境で低コストを実現するための Hybrid-P2P アーキテクチャには、オブジェクト密集状態での性能低下という問題点がある。そこで本稿ではパーティーという概念を導入し、パーティ Leader を用いた負荷分散によって、密集状態にも対応した負荷分散法を提案した。そして、シミュレーションを行った結果、密集状態でも性能が落ちていない事が確認できた。

本研究の一部はグローバル COE プログラム「アクセス空間支援基盤技術の高度国際連携」により行われました。

### 参考文献

[1] Alvin Chen, Richard R. Muntz, "Peer Clustering: A Hybrid Approach to Distributed Virtual Environments", ACM Netgames'06, 30-31, 2006.