

P2P 型オンライン RPG における動的領域分割法

榎原博之*

Dynamic Region Decomposition Method for P2P Based Online RPG

Hiroyuki EBARA

1 はじめに

近年、一般家庭への高速常時接続回線の急速な普及によって、大人数がネットワークを介して同時に大規模仮想空間内で遊ぶ多人数参加型オンラインゲーム (MMOG: Massively Multiplayer Online Game) が注目されている。1970年代後半に CAI (Computer Aided Instruction) システムなどの応用によって、MMOG の基礎となる複数人が同じ仮想空間内でプレイできるゲームがアメリカで開発された。1990年代中盤以降の爆発的なインターネットの普及に伴い MMOG は発展し、数人から数百人で遊ぶシューティングゲームやパズルゲーム、ロールプレイングゲームなどの様々なソフトウェアが登場した。その中でも、オンライン RPG (MMORPG: Massively Multiplayer Online Roll Playing Game) は圧倒的な人気を誇り、最も加入者数の多い MMORPG である World of Warcraft^[2] は、一時は全世界加入者数が1200万人にも及んだ。

MMOG は通信接続方式によって「C/S (クライアント・サーバ) 型」と「P2P (Peer to Peer) 型」の2種類に分類することができる。現在の MMOG では一般的に C/S 型の通信接続方式が用いられている。この方式は、中央管理サーバにてゲームサービスを提供し、クライアントとなるユーザがサーバに接続することでゲームを進行する。中央管理サーバがセキュリティの確保やデータの管理を一括して行うので、コンテンツの管理が容易にできるという利点がある。しかし、C/S型は全ての負荷がサーバに集中するため、サー

バの処理能力を超える負荷が発生したとき、最悪の場合サービスが一時停止してしまうスケーラビリティの問題がある。この問題は処理能力の高いサーバや広帯域ネットワークを確保することで補うことができるが、初期費用やネットワーク回線の維持費などが必要となりコストがかさむ。

そこで、C/S 型の通信方式にかわって P2P 型の通信接続方式が検討されている。P2P 型 MMOG では、大規模なゲーム領域を複数の部分領域に分割し、部分領域毎にサーバの役割となる担当ノード (以下、管理ノード) をプレイヤーの中から決定し、局所的な C/S 型を構築してゲームを進行する。サーバを介さずにピア同士で負荷を分散するため、上記の C/S 型特有の問題を解決できる。しかし、ゲームを進める上でピア接続切れ時のデータ保護、ピア間で通信する際のデータの整合性の維持、プレイヤーの局所的な集中による管理ノードの処理能力を超える負荷の発生といった問題がある。このような問題を解消するため、今日までにいくつかの領域分割手法^{[4][5][8]}や管理手法^{[1][3][6][7]}が検討、提案されてきている。

本研究では、MMOG の中でも多人数が同時にプレイする P2P 型 MMORPG における動的な領域分割手法を提案する。P2P 型 MMORPG は、「固定分割手法」や「四分木手法^[4]」といった領域分割手法が既に提案されているが、本研究では動的に領域を分割する手法を提案する。本提案法は、各部分領域に部分領域内に存在できるプレイヤー数の上限1と下限の2種類の閾値を設けることでプレイヤーの移動を考慮した動的な領域分割・結合を行う。本提案を用いることで既知の手法よりも一層の負荷低減を図る。

原稿受付 平成24年9月18日

*システム理工学部 電気電子情報工学科 准教授

2 領域分割手法

2.1 管理ノード

P2P型MMORPGでは、大規模なゲーム領域を複数の部分領域に分割しゲームを進行する。この領域分割してできた部分領域を管理するピアのことを管理ノードと呼ぶ。管理ノードは部分領域ごとに設けられ、設けられた部分領域内でサーバの役割を果たす。管理ノードは一般ピアから選ばれ、一般のプレイヤーと同じように移動などを行う。従って、管理ノードは自身が管理している部分領域に滞在している必要はなく、別の部分領域に属する場合もある。管理ノードの役割は以下の通りである。相互干渉領域については後で説明する。

- 自身が管理する部分領域内のプレイヤー数が大きくなると、領域を分割し新たな管理ノードを決定する。
- 領域分割後、各部分領域の管理ノードは自分の領域のデータをその領域に属する全プレイヤーに送信する。
- 部分領域内に存在するプレイヤーと隣接する領域の管理ノードの座標データを取得する。
- 移動によってプレイヤーが相互干渉領域に属した場合、プレイヤーに相互干渉領域に隣接する管理ノードの座標データを送信する。

管理ノードは計算能力が高く、通信帯域に余裕のあるノードが選定されることが望ましい。また、分割後も同じ管理ノードを引き続き採用すると、管理ノードが領域分割後に部分領域のデータを送信する負荷が減少する。

2.2 相互干渉領域

P2P型では、大規模仮想空間を部分領域に分割し、各部分領域に管理ノードを割り当て管理することでプレイヤーが仮想空間内を自由に移動できるようにしている。しかし、ただ領域を分割するだけでは、プレイヤーが管理されている部分領域から違う部分領域へ移動する際にリアルタイムな情報更新ができず、データの誤差が生じてしまう場合がある。そこで、管理ノード間のプレイヤーデータを受け渡す場所として相互干渉領域を部分領域と部分領域の間に設置し、データの整合性を維持する。この相互干渉領域を設けることで、管理ノード間で整合性のとれた情報更新が可能となる。

2.3 動的領域分割手法

領域分割はプレイヤーの移動、部分領域内のプレイヤー

数を考慮して動的に行う必要がある。そこで、分割閾値、結合閾値、領域プレイヤー数の3つの値を設定する。これらの値を用いることで、プレイヤー数に対応して領域数を増減する動的な分割手法を提案する。以下に、分割閾値、結合閾値、領域プレイヤー数について説明する。

• 分割閾値

任意に決めた各部分領域内に存在できるプレイヤー数の上限値を示す。

• 結合閾値

任意に決めた各部分領域内に存在できるプレイヤー数の下限値を示す。

• 領域プレイヤー数

領域を分割する際に各部分領域が内包するプレイヤー数を示す。分割閾値と結合閾値の平均の切り上げ値とする。

以降、領域分割手法やアルゴリズムを説明する際に用いる図では、分割閾値5、結合閾値1、領域プレイヤー数3と仮定する。また、仮想空間は2次元平面であり、横軸(x軸)と縦軸(y軸)からなるものとし、マス目単位で領域を扱うものとする。

動的領域分割手法の流れを以下に示す。

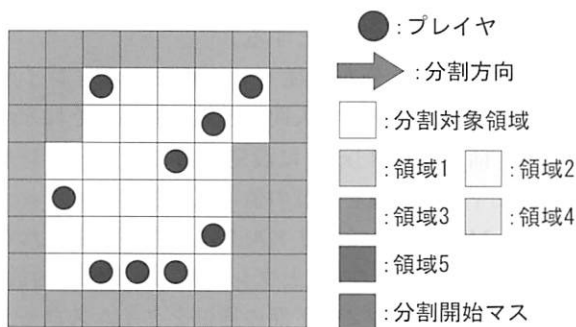
- (1) 分割閾値、結合閾値を決める。また、分割閾値と結合閾値の平均を分割プレイヤー数とする。
- (2) 部分領域内のプレイヤー数が分割閾値を超える場合、または結合閾値を下回る場合、その領域と接する領域の全てを統合し1つの領域(以下、分割対象領域)とする。
- (3) 分割対象領域に対して分割判定(後述)により、分割を開始するマス(以下、分割開始マス)と分割を進める方向(以下、分割方向)を決める。
- (4) 分割判定で決定した分割開始マスと分割方向を基に、分割対象領域に対して領域分割(後述)を行う。
- (5) 内包するプレイヤー数が統合領域を下回る、または部分領域の縦横の長さの比が1:2あるいは2:1以上の部分領域に対して領域修正(後述)を行う。
- (6) 新しく作成された部分領域に管理ノードを割り当てる。
- (7) 分割閾値を超える、または結合閾値を下回る度、(2)~(6)を繰り返し行う。

2.3.1 分割判定

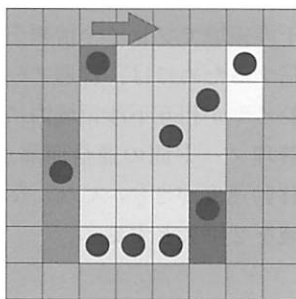
分割対象領域の形によっては、分割すると細長い部分領域ができやすいものがある。細長い部分領域は、

プレイヤーが移動する際に異なる管理ノードが管理する領域を跨ぐ頻度を増やすので、管理ノード間の通信負荷の増加に起因するという問題点がある。そこで、分割対象領域の形によって分割開始マスを変更し、分割する際の部分領域を広げていく順番も変えることで、ある程度細長い領域をできにくくすることができる。なお、x 長、y 長の順に部分領域を広げ、x 軸方向に分割を進めていくことを横方向とし、y 長、x 長の順に部分領域を広げ、y 軸方向に分割を進めていくことを縦方向とする。分割開始マスと分割方向の違いによる領域分割後の部分領域の違いを図 1 で示す。

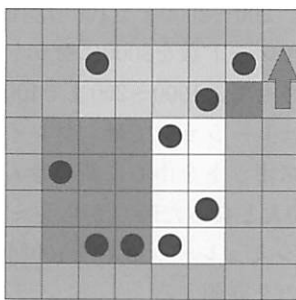
図 1 より、分割の開始マスと分割方向によって同じ形の分割対象領域でも、分割後の部分領域の形が大きく異なることがわかる。以下に分割判定のアルゴリズムを示す。



(a) 分割前



(b) 最も上の左から横方向に領域分割を行った場合



(c) 最も右の下から縦方向に領域分割を行った場合

図 1 分割開始マスと分割方向

- (1) 分割対象領域の範囲を調べる。
- (2) 分割対象領域を全て内包する最小の四角形を作成する。
- (3) 作成した四角形から分割対象領域分を削除する。
- (4) 残ったそれぞれの領域の x 長と y 長の差を比較し、一番差の大きい領域 (領域 A と仮定) を選択する。
- (5) 領域 A の x 長、y 長を比べ、短い辺と接する分割対象領域の角のマスを分割開始マスとする。また分割方向は、x 長が y 長以上の場合は横方向、y 長の方が大きい場合は縦方向とする。この例では、上に進む。

2.3.2 領域分割

分割対象領域に対して領域分割を行う。領域プレイヤー数を超えない範囲で x 長、y 長を交互に増加させる。また、部分領域の x 長を増加中に他の部分領域に接した場合、x 長の増加を停止し y 長のみを増加させ部分領域の範囲を広げる。同様に、部分領域の y 長を増加中に他の部分領域に接した場合、y 長の増加を停止し x 長のみを増加させ部分領域の範囲を広げる。このような条件を加えることで、部分領域を必ず四角形に保つことができる。

- (1) 分割対象領域の有無を調べる。
- (2) 分割対象領域の左上から分割を開始する。
- (3) 部分領域の x 長を 1 マス増加させる。
- (4) 部分領域の y 長を 1 マス増加させる。
- (5) 部分領域が内包するプレイヤー数が領域プレイヤー数を超えない、または他の領域と接さない限り (3)、(4) の動作を繰り返す。
- (6) 仮に x 軸で部分領域同士が接した場合は x 長の増加を停止し、y 長のみを内包するプレイヤー数が領域プレイヤー数を超えない、または他の領域と接さない限り増加させる。y 軸が接した場合も同様である。
- (7) 分割対象領域が完全に分割されるまで、(2)~(6) を繰り返す。

2.3.3 領域矯正

本提案手法は内包するプレイヤー数に加えて部分領域の形も考慮するため、場合によっては内包するプレイヤー数が結合閾値を下回る部分領域や、細長い部分領域ができてしまう。そこで、結合閾値を下回る領域と、部分領域の x 長と y 長の比が 1 : 2 あるいは 2 : 1 以上の部分領域を矯正する領域矯正を行う。以下に、領域矯正の流れを示す。

- (1) 結合閾値を下回る領域、あるいは部分領域の x 長と y 長の比が 1 : 2、または 2 : 1 以上の部

分領域を検知する。

- (2) 統合する領域によっては、統合後の領域の形が六角形になることがある。そこで、検知した部分領域が縦長の場合は、統合後にできる六角形の領域数が少ない、左右のどちらかの領域と統合する。統合後にできる六角形の数が同じ場合は左の領域と統合する。同様に横長は、上下の六角形の数を比べ、少ない方と統合する。統合後にできる六角形の数が同じ場合は上の領域と統合する。x 長、y 長が等しい場合は、上下左右の統合後の六角形のできる数をそれぞれ調べ、最も少ない方向と統合する。六角形の数が同じ場合は左の領域と統合する。
- (3) 統合後の部分領域が内包するプレイヤー数が分割閾値より多い場合、その部分領域の形が縦長か横長かを調べる。
- (4) 横長の場合、部分領域が内包するプレイヤー数が半分ずつになるように領域を縦に分割する。同様に、縦長なら部分領域が内包するプレイヤー数が半分ずつになるように領域を横に分割する。
- (5) 検知した全ての部分領域に対して(2)～(4)を行う。

3 性能評価

提案法の有効性を示すため、コンピュータシミュレーションによる性能評価を行う。シミュレーションでは、四分木手法（以下、四分木）による領域分割法と、提案法をそれぞれ比較検討し、それらの性能について評価する。

3.1 シミュレーションモデル

シミュレーションに際し、仮想空間、閾値、プレイヤー、相互干渉領域、通信間隔、負荷の設定を以下に示す。

1. 仮想空間

仮想空間は 64×64 マスの正方形の2次元平面とし、領域の最小単位1マスとする。また、現実世界で人が不快なく存在するためには、ある程度空間が必要である。その快適な空間のことをパーソナル・スペースとよび、一般的に $45 \sim 75\text{cm}$ とされている。よって仮想空間内の1マスは 60cm と定義する。

2. 閾値

分割閾値を30、結合閾値を10、領域プレイヤー数を20とする。

3. プレイヤ

- MMORPGでは通常、グループで集まるため、

プレイヤーは一様に分布しにくい。このことから、プレイヤーの振る舞いを完全なランダムではなく、ある程度の動作パターンを持たせるために、プレイヤーノードの新規参入及び、移動が集中する部分（以下、ホットスポット）を5ヶ所設定する。

- タイムスロット（1[s]）毎に移動できる範囲は、人間の歩く速度が秒速約 1m なので2マス（ 120cm ）とする。また、プレイヤーの移動は、確率に従い上下左右の移動または移動しないの5パターンの行動をランダムに2回行う。各々の行動の確率は $2/11$ である。また、 $1/11$ の確率であらかじめプレイヤーに決められたホットスポットへ向かう方向に移動するようにする。
- プレイヤの参入及び離脱は、増加時と減少時でデータに関係性を持たせるために、増加時は毎秒2人参入し、1人離脱、減少時は参入無しの1人離脱とする。
- ポワソン分布に従って、離脱するプレイヤーノードの新規参入間隔と離脱間隔をそれぞれ同じ値（3[s]）に設定することで、プレイヤーノード数を任意の値に収束させている。
- MMORPGではリアルタイム性が重視されるので、管理ノードとプレイヤーノード間の通信、および管理ノード間の通信をタイムスロット毎に行うものとする。

4. 相互干渉領域

相互干渉領域でプレイヤー情報を管理ノード間で通信するためには、管理ノード間の通信中にプレイヤーが相互干渉領域内にいる必要がある。管理ノード間の通信間隔が1[s]、プレイヤー移動速度が $2[\text{マス}/\text{s}]$ なので、相互干渉領域を2マスとする。

3.2 シミュレーション結果

プレイヤーノード数を100にセットした後、毎秒1ずつ増加させて、200～2000まで100増加毎に測定する。また、プレイヤーノード数を2000にセットした後、毎秒1人ずつ減少させて、1900～200まで100減少毎に測定する。各シミュレーションでは、プレイヤーノードは設定したホットスポットを中心に新規参入し、プレイヤーノード数が100人となった時点からシミュレーションを開始する。シミュレーションを100回を行い、その平均をとったものを評価する。

まず、各手法におけるプレイヤー数の増減に対する領域数の変化を図2、3に示す。図2より、提案分割手

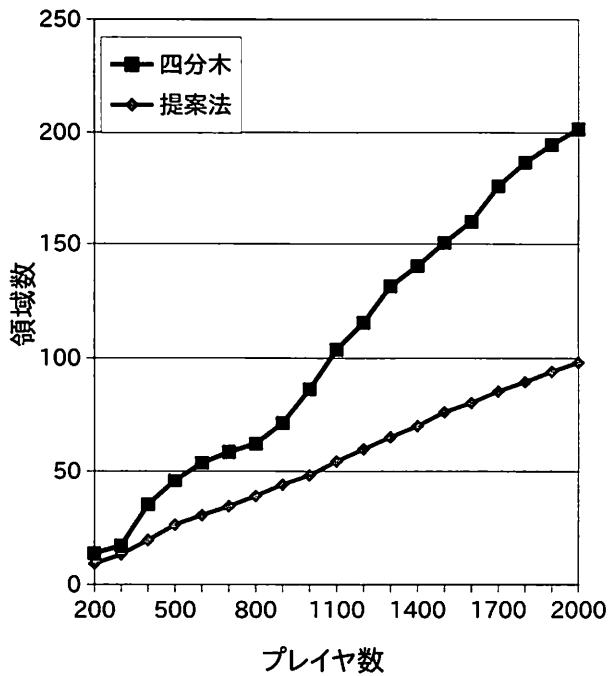


図2 領域数 (プレイヤー増加時)

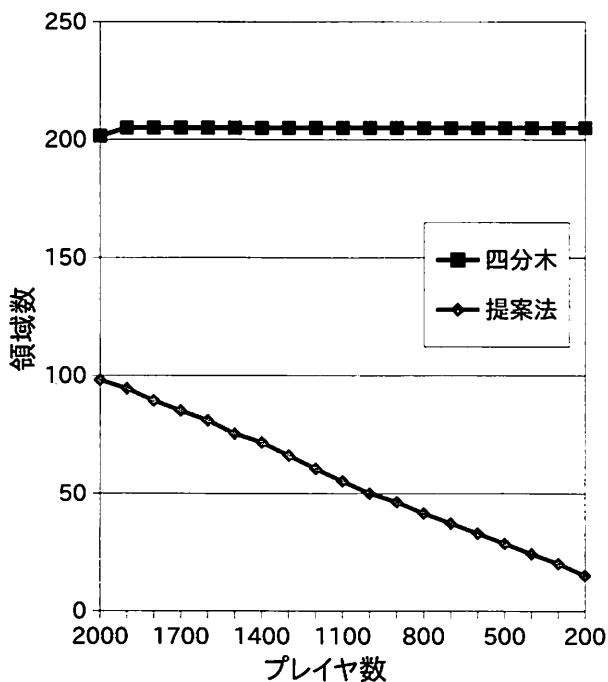


図3 領域数 (プレイヤー減少時)

法は、四分木に対して最大半分程度まで領域数を削減することに成功している。また、図3より、四分木はプレイヤー数が減少しても小さくなった領域を統合できないために、領域数を削減できていないことがわかる。それに対して、提案分割手法は無駄な領域の削減や小さくなった領域の統合が可能のため、プレイヤーの減少に対応することができている。次に、プレイヤー数に対する総負荷を図4、5に示す。図4、5より、提案法

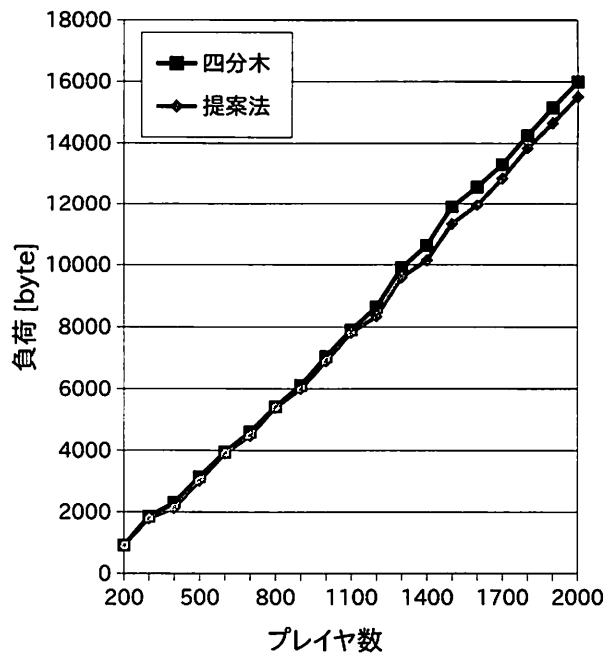


図4 総負荷 (プレイヤー増加時)

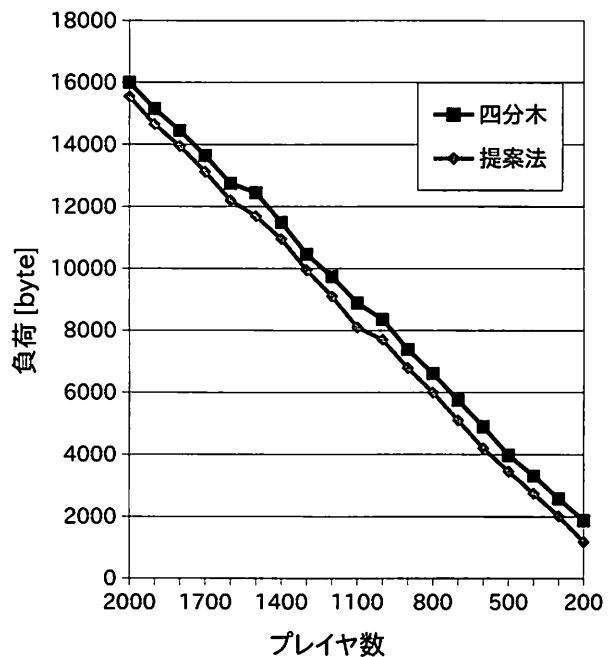


図5 総負荷 (プレイヤー減少時)

は四分木に比べ総負荷を抑えられることがわかる。

4 おわりに

本研究では、大規模仮想空間における動的領域分割手法を提案した。また、シミュレーションにユーザの密集が起りやすいホットスポットや、データの整合性を維持するために相互干渉領域を導入することで、より実際のMMORPGに近い環境でシミュレーションし、どの程度負荷を軽減できるかを評価した。また、

提案法は、動的な領域分割を行うだけでなく、歪な形の領域を少なくすることにより負荷の軽減ができることを示した。結果より、プレイヤーの移動を考慮した動的な領域分割をすることで、プレイヤーの密集や過疎に対応した負荷分散ができることを示した。

現在、管理ノードは離脱しないものとして、シミュレーション実験を行っている。管理ノードの離脱に関しては、データの損失を避けるため、バックアップを保存するプレイヤーを選ばなければならない。管理ノードの離脱は今後の課題である。

参考文献

- [1] Shervin Shirmohammadi Dewan Tanvir Ahmed. A dynamic area of interest management and collaboration model for p2p. *Proceedings of the 2008 12th IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications*, pp.27-34, 2008.
- [2] Blizzard Entertainment. World of warcraft. (<http://www.worldofwarcraft.com/index.xml/>)
- [3] Velin Tzanov Marios Assiotis. A distributed architecture for mmorpg. *NetGames '06 Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*, pp.73-78, 2006.
- [4] Ariel Vardi Philippe David. Improving scalability in mmogs – scalamo: a new architecture – , 2006.
- [5] M. Kurashige R. Thawonmas and K.T. Chen. Detection of landmarks for clustering of online-game players. *International Journal of Virtual Reality*, Vol.6, No.3, pp.11-16, 2007.
- [6] 小林基成, 鈴木俊博, 永田智大, カーンアシック, 趙晩熙. オンラインゲームのための仮想ネットワークの資源割り当て方法の検討. 電子情報通信学会技術研究報告. NS, ネットワークシステム, pp.73-78, 2008.
- [7] 山崎孝裕, 金田憲二, 大山恵弘, 米澤明憲. 柔軟性と拡張性を備えた大規模多人数オンラインゲームのための枠組み. SWOPP 武雄2005, pp.61-66, 2005.
- [8] 遠藤伶, 高木健士, 北望, 重野寛. Mmo 仮想環境におけるユーザ密度の変化に対応した負荷分散手法 (セッション b-8:p2p・オーバーレイネットワーク (2)). 情報処理学会研究報告 (CSEC), No.21, pp.213-218, 2008.