M-082

P2P 環境でのネットワークゲーム向け負荷分散機構と評価
A Load Balancing Mechanism for Multi-player Games on P2P Networks and its Evaluation

山本 眞也 * Shinya Yamamoto　**安本 慶一** * Keiichi Yasumoto　**村田 佳洋** * Yoshihiro Murata　**伊藤 実** * Minoru Ito

1. まえがき

近年，ピアツーピア (P2P) の環境で特定のサーバを設置することなく多人数参加型ネットワークゲームを実現する方法が注目されている．従来，ゲーム空間を均等に分割し，分割されてできた複数の領域におけるイベント通知を，固定数のサーバに対応させる方式 [1] や， P2P 環境でいくつかのプレイヤの端末に割り当て処理する方式 [2] が提案されている．しかし，これら従来の方式を P2P 環境にそのまま適用した場合，ある領域のプレイヤ数およびそこで発生するイベントの数が多くなると，その領域を担当するノードに負荷が集中するという問題があった．我々は，文献 [3] において，均等分割した各領域のプレイヤ数とそこで発生するイベントの発生頻度を監視し，その領域でのイベント通知を担当するノードの負荷が予め定めた水準を超えないよう，領域内のプレイヤの集合を部分集合に動的に分割し，新たに割り当てたノードに分散処理させる方式を提案している．本稿では，階層ネットワーク上に P2P ネットワークを構築し，イベント配送にかかるエンドツーエンド遅延時間，および，必要帯域を予測することで，本方式の有効性を確認する．

2. イベント通知機構と負荷分散の概要

我々が文献 [3] で提案したイベント通知機構（以下，本方式）では，ネットワークで接続された複数のプレイヤーが同一の仮想空間と時間を共有する，鳥瞰型のロールプレイングゲームを想定している．仮想空間は，背景と複数の**オブジェクト**からなるものと定義する．ここで，オブジェクトは，プレイヤーや他の移動体および静止物であり，それぞれ，属性を持つ．オブジェクトの属性を変化させる出来事を**イベント**と呼ぶ．イベントの例として，ユーザの移動や，他のオブジェクトへの働きかけ，オブジェクトの色・形状の変化などがある．このようなネットワークゲームにおいて，複数プレイヤによるゲーム進行の一貫性を保証するには，以下を満たす必要がある．
(1) 仮想空間上のある範囲にいる全てのプレイヤーは，常に同じゲーム状態（領域内に存在する全てのオブジェクトの位置とその属性）を保持する．
(2) その範囲で発生したイベントは，範囲内の全プレイヤーにリアルタイムで伝達される．
(3) 連続して発生した一連のイベントの発生順序が，同一範囲内の全プレイヤーで一意に保たれる．

本方式は，ピアツーピア環境において上記を実現するため，以下の方針を採用している．まず，プレイヤのリストを保持するロビーサーバ S の存在を仮定する．発生したイベントの影響範囲をプレイヤから見える範囲（**可視領域**と呼ぶ）に限定し，仮想空間を均等な M 個の可視領域に分割する．(図 1) そして，それらの可視領域を担当する，ある基準で選ばれた M 個のノード (以後，担当ノードと呼ぶ) で，リング型トポロジーを構成する．そ

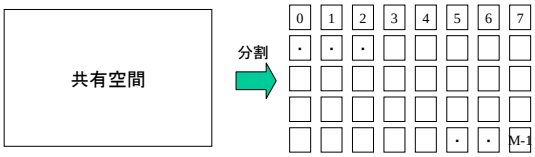

図 1: 仮想空間の分割

れらに Chord [4] による分散ハッシュテーブルおよび隣接領域へのショートカットを持たせることで，ユーザが異なる領域に移動した時のイベント転送（または購読予約の転送）にかかる遅延時間を短縮している．

担当ノード以外のノードは，publish / subscribe モデルを用いてイベントの配送要求をイベント担当ノードに購読予約 (subscribe) し，可視領域および近隣領域で発生したイベントのみを受信することにより通信量の削減をはかる．また，各可視領域での負荷 (subscribe しているプレイヤ数 × 単位時間あたりに発生するイベント数) が予め指定した閾値 C を越えた場合には，リング上の該当ノードを根とする木構造のネットワーク (以後，**負荷分散木**と呼ぶ) を動的に構築することで負荷分散を実現する．(図 2)

リング上のノード A に subscribe しているプレイヤ (subscriber) 群を分割し，k 分木の複数のノード (ロビーサーバにより選択されたノード) に分散して担当させることにより，単位時間あたりに処理すべきイベントの数を削減する（ここで，k は予め与えられる木の分岐数とする）．リング上の各ノードは，あらかじめロビーサーバから，子ノードの候補となるプレイヤノードの情報を受けとっていると仮定する（負荷の低い領域の担当ノードを候補としても良い）．リング上のノード A は，それらのノードから k 個を選び，子ノードとして接続する（“ping” などを使って遅延時間が最小となるものから選んでも良い）．ノード A は，以下の手順で負荷分散木を構築し，子ノードへ subscriber の割り当てを行う．
(1) ノード A は，現在の subscriber の集合を，ほぼ同数になるよう k 分割し，分割された k 個の subscriber の集合を，A に接続された k 個の子ノードに送信する．

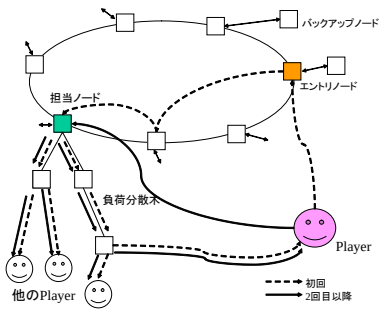

図 2: イベント通知機構と負荷分散

†奈良先端科学技術大学院大学，NAIST

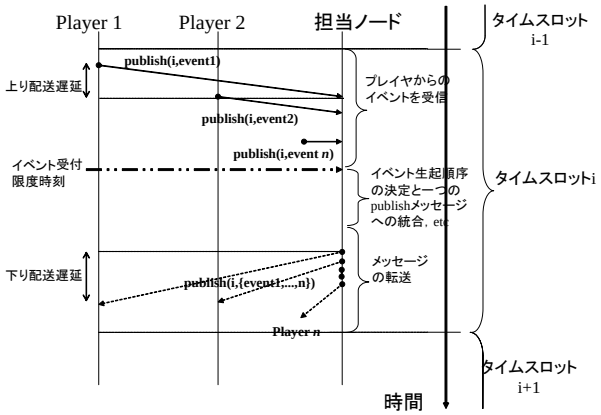

図 3: タイムスロット

(2) ノード A が publish メッセージを受け取ったら，k 個の子ノードに転送する．また，A が subscribe メッセージを受け取ったら，子ノードのうち，最も subscriber が少ないノードに転送する．
(3) 負荷分散木のある葉ノードの負荷が高くなったら（subscriber の数×単位時間あたりに発生するイベント数が閾値 C を越えたら），手順 (1) に従い，subscriber の集合を分割し，子ノードに割り当てることで再帰的に負荷を分散する．
(4) 負荷分散木のある中間ノードを根とする部分木の subscriber の数×単位時間あたりのイベント数が閾値 C 以下になったら，子ノードの統合を行う．そのため，各中間ノードに，イベントの発生頻度，子ノードの担当する subscriber の集合をモニタさせる．

ゲーム空間および時間の流れに一貫性を持たせるために，各ノードの時計は NTP を用いてある程度の正確さを持たせる．図 3 に示すようなタイムスロットによる同期機構を設け，各スロットで発生したイベントを集約し 1 つのメッセージとして配送することで，プレイヤー間でのイベントの生起順序を一意に保つ．タイムスロットは，メッセージに付加されているタイムスロット番号と現在のタイムスロット番号を比較し，一致する場合のみ受理し，そうでない場合は破棄する．イベント配送メッセージが失われた時は，タイムスロット番号による NACK メッセージにより，イベントの再送を依頼する．

3. 実験と評価

一般に，ゲームで遊ぶユーザは自分の入力が反映されるまでに 400ms 以上経過すると「結果が反映されていない」と感じることが知られている [5]．

そこで，階層型ネットワークに本手法を適用した場合の，イベント通知にかかるエンドツーエンド遅延，イベント配送にかかる帯域を簡単な解析により調べた．

WAN に接続されている MAN の数を 5，各 MAN に接続されている LAN の数を 5，各 LAN に接続されているノード数が 12 であるような階層型ネットワークを想定する．なお，同一 LAN のノード間の遅延時間を 1ms，MAN を経由する LAN ノード間の通信の遅延を 2ms，WAN を経由する LAN ノード間の通信遅延を 15ms と考える．総ノード数は 300 となる．ある領域のプレイヤ数を 45，各プレイヤのイベント発生頻度を $e = 5/sec$ とし（すなわち，タイムスロットは 200ms），閾値 $C = 5 \times 5 = 25$，木の分岐数は $k = 3$ とする．この場合，負荷分散木の段数は 2 段になる．

プレイヤ，および，負荷分散木を構成するノードは，LAN に接続されているノードからランダムに選択し，パケットサイズは，プレイヤが生成する各イベントのペイロード 50 バイトと各階層のプロトコルヘッダ 50 バイトの和となる 100 バイトとした．領域の担当ノード（リング上のノード）は，各タイムスロットで受け取ったイベントを集約し転送するため，担当ノードが転送する publish メッセージのペイロードは，50*45=2250 バイトとなる．2 つのパケットで送信する場合，ヘッダを含めて，各タイムスロット毎，2350 バイトの情報が負荷分散木を根から葉に向かって転送されることになる．以上の設定で，イベント配送にかかるエンドツーエンド遅延の平均と最大，使用帯域のリンクごとの平均と最大を見積もることにする．

イベントは，プレイヤノード→負荷分散木の根ノード→節ノード→葉ノード→プレイヤノードの順で配送される．このため，根ノードでのメッセージ集約の計算時間を無視した場合，エンドツーエンド遅延は，全ての通信が WAN を経由する場合が最悪となり，4*15=60ms となる．実際には，各ノードでパケットを中継するための遅延が生じるが，200ms の制約を十分満たせると思われる．

一方，負荷分散木の根ノードは，200ms 毎に，45 人のプレーヤからイベントを受け取り，集約後のメッセージを 3 個の節ノードに配送する．従って，根ノード付近では，(45*100+3*2250)/200ms=101.25KB/s= 810Kbps の帯域が必要となる．節ノードでは，200ms 毎に 1 個の集約メッセージを受け取り，3 個の子ノードへ転送を行う．従って，必要な帯域は，2250*4/200ms=45KB/s=360Kbps の帯域が必要となる．プレイヤノードでは，200ms 毎に，1 個のイベント情報を送信し，1 個の集約メッセージを受け取る．従って，必要な帯域は (100+2250)/200ms =94Kbps となる．

以上より，負荷分散木を構成するノードとして，比較的広帯域が利用可能なノードを選べば，提案手法が実用的に利用可能なことが分かった．

4. まとめ

本稿では，ネットワークゲームを P2P ネットワーク上でサーバレスで実現する際の負荷分散機構とその評価について述べた．階層ネットワークを想定した性能解析により，本手法が，広域ネットワークでのソフトリアルタイム RPG の実現に十分適用可能なことを確かめた．今後，ns-2 上に本手法を実装し，より詳細な評価実験を行う予定である．

参考文献

[1] 井芹，堀，藤川，下條，宮原：多人数参加型ネットワークアプリケーションの広域ネットワーク環境における利用実験，信学技法，IN2000-121，pp. 21–28 (2000)．

[2] A. Bharambe, S. Rao and S. Seshan: “Mercury: A Scalable Publish-Subscribe System for Internet Games”, *Proc. of 1st Workshop on Network and System Support for Games (NetGames2002)* (2002).

[3] 公原勝彦，安本慶一，伊藤実：P2P 環境でのネットワークゲーム向け負荷分散機構の提案，第 11 回マルチメディア通信と分散処理 (DPS) ワークショップ論文集，pp. 79-84 (2003)

[4] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek and H. Balakrishnan: “Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications”, *Proc. of ACM SIGCOMM’01*, pp. 149–160 (2001).

[5] T. Henderson: “Latency and Behaviour on a Multiplayer Game Server”, *Proc. of 3rd Intl. Workshop on Networked Group Communication (NGC2001), LNCS2233*, pp. 1–13 (2001).