

リアルタイムグループウェアにおける 途中参加方式についての一考察

五郎丸秀樹 関野公彦 久保田創一 佐藤栄
NTT 情報通信研究所

概要

近年、不特定多数のユーザがグループを構成し、ネットワーク上のリアルタイムアプリケーションを共有する形態が増加している。本形態の多くのアプリケーションではユーザが途中からグループに参加できる途中参加機能を設けることが予想される。しかし、途中参加の際に既存参加者に対するサービスが中断することや参加希望者の参加終了までの時間がかかるといった様々な問題が発生することが考えられる。本稿では途中参加の方式の検討、特に多くのユーザが参加している状態で途中参加を行なう上での問題点を指摘し、新たに考案した擬似クライアントを設けることにより参加時間が短縮されることを述べる。

An entry method for real-time groupware

Hideki Goroumaru Kimihiko Sekino Souichi Kubota Sakae Sato
NTT Information and Communication Systems Laboratories

Abstract

The number of users forming various working groups and sharing real-time applications on a network is rapidly increasing. Many such applications allow a user to join a working group at any time, but service can be disrupted and the joining process can take a long time. Having researched the methods of joining working groups, we propose the idea of the pseudo client. The pseudo client reduces the time needed to join or leave a group.

1. はじめに

今日、ネットワークのオープン化により不特定多数のユーザがネットワークに接続するようになった。ユーザがネットワーク上の共有アプリケーションを使用してグループを形成し協調作業を行なう機会^[1]が多くなるにつれて、他のユーザがそのグループに途中から参加する形態も増えてくることが予想される。そのためグループ内のユーザ同士の協調作業が終了するまで待たずに他のユーザがそのグループに参加することを可能にする途中参加機能が必要となってくる。

2. 対象モデル

ネットワークのオープン化で1000人以上の人数で協調作業を行なう多人数参加型のアプリケーションが出現する可能性がある。現在でもSIMNET^[2]では800人のユーザが参加可能である。途中参加方式について、考案するために、以下の対象モデルを想定する。

対象モデルは、イベントの制御を行なうセンター、参加者間で共通に持つデータ（共有データ）と共有アプリケーションを持つ端末から構成される(図1)。センターは既存参加者からのイベン

トを集めてマルチキャストを行なうイベント配信サーバを持つ。

各端末の共有アプリケーションは、受信したイベントに基づいて共有データを書き換える。本モデルではすべての参加者が同じイベントを受けとり、通信遅延の影響を考えなければ各参加者の共有データは等しいものとする。

途中参加は、ある既存参加者がセンターを経由して参加希望者に対して任意の時刻の共有データを送信することにより行なわれる。

そして参加希望者が共有データと共有アプリケーションを持ち、既存参加者と同じ状態になるとき参加終了と見なす。

対象モデルで使用されている用語について説明する。

既存参加者とは既に共有データと共有アプリケーションを持ち協調作業を行なっている端末を表す。参加希望者とは共有アプリケーションは持つが、共有データを持たないか持っても既存参加者との同期がとれておらず協調作業を行えない状態の端末を表す。共有データとは参加者が様々な協調作業を行なうための同一のデータである。

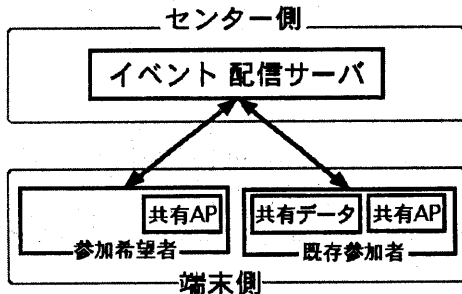


図1 対象モデル

3. 途中参加方式について

途中参加の方式は、途中参加処理中は既存参加者へのサービスの提供を一時中断させる「中断方式」と既存参加者へのサービスを提供し続ける「継続方式」の二つに大きく分けられる。ここでは、この両方式からそれぞれの特徴を述べ、途中参加のフローを作成し、参加処理時間を表すモデル式を作成する。

3.1. 中断方式

中断方式は、参加希望者が途中参加を要求した時から途中参加が終了するまでの間、既存参加者へのサービスを一時中断させる方式である。

中断方式による途中参加のフローを図2に示す。このフローは単純化した途中参加処理時間を求めるために作成した。このフローと表1、表2に使われている用語について説明する。ノード間イベント送信時間とはセンターと端末の間をイベントが通過するときにかかる時間である。停止要求とは、各端末に協調作業に関するイベントの送信を停止してもらうようにセンターが各端末に停止を促すことである。送信用共有データ作成時間とは参加希望者に送信する任意の時刻の共有データを作成する時間のことである。共有データ送信時間とは既存参加者の共有データの送信の開始から参加希望者が共通データの受信を終了するまでの時間である。

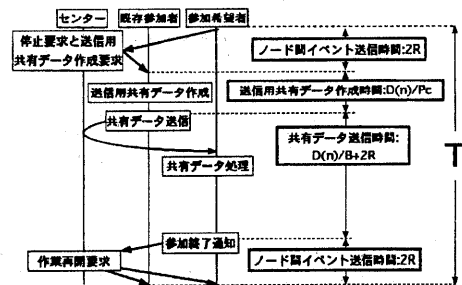


図2 中断方式の途中参加のフロー

また途中参加処理時間式の簡素化のために前提条件として、「各ノード間のイベント送信時間は全て等しく、ノード間イベント送信時間はイベントのノード間の経路を通ることによる通信遅延に近似させる」「“停止要求と共有データ作成要求”“参加終了通知”“作業再開要求”を行う時間は無視する」「送信用共有データ作成時間は共有データのデータ量に比例する」「共有データ送信時間は、共有データのデータ量に関する一次関数で近似する」「計算機処理能力は、アプリケーションやOS、メモリ等のソフトやハードに依存するが、ここではmemory-to-memoryであるバスのスループットに近似する」と仮定する。

中断方式の途中参加のフロー(図2)と表1、表2より参加処理時間Tを求めると

$$T=6R+D(n)/Pc+D(n)/B \quad (1)$$

となる。

表 1 中断方式に使われる式の項目

略記	項目
D(n)	n 人当たりの共有データのデータ量 $D(n)=D_h \times n + D_m$ (Byte)

表 2 中断方式に使われる変数の項目

略記	項目
Pc	端末の計算機処理能力(Byte/sec)
n	既存参加者数
B	通信路帯域幅 (bps)
R	ノード間の経路長による通信遅延(msec)
Dm	人数に左右されない共有データの固定データ量 (Byte)
Dh	人数に比例する共有データの可変データ量(Byte)

3.2. 継続方式

継続方式は、途中参加処理中であっても既存参加者へのサービスを中断させない方式である。継続方式の途中参加のフローに示す(図3)。

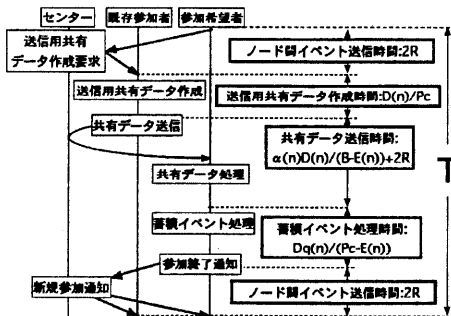


図 3 継続方式の途中参加のフロー

図3と表3,表4の用語について説明する。蓄積イベント処理時間とは、参加希望者の共有データの受信完了から参加終了通知送信までの時間であり、その間に参加希望者はキューに蓄積されたイベントを共通データに反映させることで新規参加者となる。センターのイベント伝送係数とは、センターを経由してイベントをマルチキャストする際、単位時間当たりのイベント伝送量によるセンター経由の通信路のスループットの変化を近似するための係数である。

また中断方式の場合と同様に

- 共有データ送信時間は、共有データのデータ量等の関数式で近似可能とする
- 新規参加通知を行なう時間は無視する

- 協調作業の継続のため、通信路の帯域の優先度は、共有データ送信よりもイベント送信を高くする

と仮定する。

継続方式の途中参加のフロー(図3)と表(3),表(4)より参加処理時間Tは

$$T = 6R + D(n)/Pc + \alpha(n)D(n)/(B-E(n)) + Dq(n)/(Pc-E(n)) \quad (2)$$

となる。

表 3 途中参加に使われる式の項目

略記	項目
D(n)	n 人当たりの共有データのデータ量 $D(n)=D_h \times n + D_m$ (Byte)
E(n)	単位時間当たりのイベント平均伝送量 $E(n)=n(D_e + H_d) \times F$ (bps)
$\alpha(n)$	センターのイベント伝送係数 $\alpha(n)=B/(B-E(n))$
Dq(n)	参加希望者のキューに溜るイベントのデータ量 $Dq(n)=(2R + \alpha(n)D(n)/(B-E(n))) \times n \times D_e \times F$ (Byte)

表 4 途中参加に使われる変数の項目

略記	項目
Ps	イベント配信サーバの計算機処理能力(Byte/sec)
Pc	端末の計算機処理能力(Byte/sec)
n	既存参加者数(msec)
B	通信路帯域幅(bps)
R	ノード間の経路長による通信遅延
De	イベントの平均データ量(Byte)
Hd	TCP/IP のヘッダ長(Byte)
F	平均イベント頻度(event/sec)
Dm	人数に左右されない固定データ(Byte)
Dh	人数に比例する可変データ(Byte)

4. 問題点

ここでは多人数参加時に生じるそれぞれの方式の問題点を指摘する。

4.1. 中断方式の問題点

中断方式は途中参加に費やす時間が増えるとサービスの中断時間が長期化するという問題がある。そこで中断時間がどの程度か検証する。

中断方式における参加処理時間の理論値を求めるため、既存のリアルタイムネットワークゲー

ム(仮想空間上でユーザの代理人であるキャラクタが移動し協調作業を行なう)を使用したときのログデータを基に求めた変数の値を表5に示す。

表5 サンプルアプリケーションでの変数の値

略記	値
Ps	320MByte/sec
Pc	133MByte/sec
n	1-1000 人
B	128k-10Mbps
R	110msec
De	21.86Byte
Hd	40 Byte
F	1.14event/sec
Dm	1015Byte
Dh	49Byte

この表5を使用して参加処理時間の理論値を求めていくと、式(1)と表5より

- 参加希望者が1人の場合
具体例として、「参加希望者数1人、既存参加者数1000人、通信路伝送容量10Mbps」の場合、参加処理時間の理論値は約700msとなる。
- 参加希望者が多数同時参加の場合
「参加希望者数10人、既存参加者数1000人、通信路伝送容量10Mbps」の場合、その累積した参加処理時間の理論値は約7sとなる。

このことからアプリケーションによっては参加希望者が集中するときは中断時間が長すぎて利用できない可能性がある。今までは参加人数が少なかったため、途中参加処理時間は無視できた。しかし中断方式では既存参加者の協調作業の中断時間が増加しリアルタイムなサービスに影響を与えることが問題となる。

4.2. 継続方式の問題点

中断方式と同様に継続方式の途中参加処理時間を表す式(2)と表5を基に継続方式の途中参加処理時間の理論値を求めた。

「参加希望者数1人、既存参加者数1000人、通信路伝送容量10Mbps」の場合、参加処理時間の理論値は734msとなる。この内訳は、

ノード間イベント送信時間 660ms 90%
共有データ送信時間 45ms 6%
イベント処理時間 29ms 4%

であり、このモデルではノード間イベント送信時間に時間がかかっているがわかる。次に「通信路伝送容量を1.5Mbps」に変えて参加処理時間を求めると1491msとなりその内訳は、

ノード間イベント送信時間 660ms 44%
共有データ送信時間 683ms 46%
イベント処理時間 148ms 10%

であり、このモデルでは共有データ送信時間に時間がかかっていることがわかる(図4)。

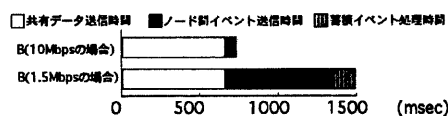


図4 通信路帯域幅の違いによる途中参加処理時間の内訳

継続方式では、通信路帯域幅が狭くなれば共有データ送信時間とイベント処理時間が参加処理時間に占める割合が高くなっていくことがわかる。

これはセンターと参加希望者間の通信路帯域の優先度は、共有データの送信よりも協調作業を行なうイベントの送信を高くしているためセンターと参加希望者間の通信路帯域が狭くなるにつれて、共有データの送信の専有する帯域が狭くなるためであると考えられる。特にセンターと参加希望者との間の通信路に大量のイベントを送らなければならない場合、途中参加に費やす時間は更に増大するであろう。

5. 提案技術

継続方式での途中参加処理時間は、通信路帯域幅が狭くなれば「共有データ送信時間」「蓄積されたイベントの処理時間」の影響が大きくなる。そこで疑似クライアントをセンターに設置することで、直接センターから参加希望者へ共有データを送信して「共通データ送信時間」を短縮させる。「蓄積されたイベントの処理時間」についてはイベントを圧縮して参加希望者へ送信することによってこれらの問題を改善した。

5.1. 擬似クライアント

擬似クライアントはセンターに置かれた仮想のプロセスであり、その目的は途中参加を行なう上で発生する諸問題を改善し参加処理時間を短縮させ途中参加を実現させることである(図5)。

その特徴は以下の通りである。

- 共有データと共有アプリケーションを持ち、直接参加希望者へ共有データを送信
- イベントの加工(蓄積、分類、圧縮、形態の変換)

である。これらの機能を用いて途中参加処理時間の短縮を図る。

5.2. 共有データ送信時間の短縮

擬似クライアントをセンターに設置することで直接共有データを参加希望者へ送信することが可能になる。そのためセンターと既存参加者間の通信路を共有データが送信される時間が短縮される。

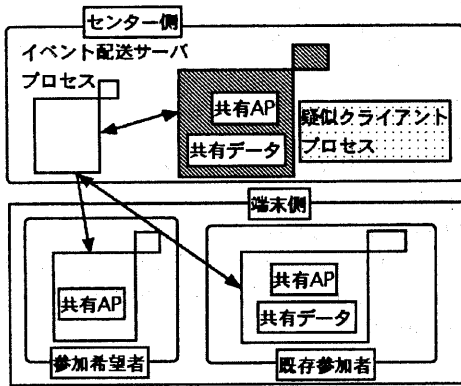


図5 擬似クライアントの構成

擬似クライアントによる途中参加のフローを図6に示す。

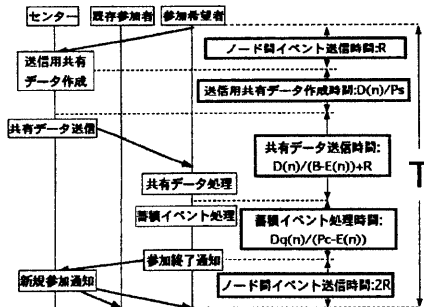


図6 擬似クライアントによる途中参加のフロー

継続方式においてセンターへ擬似クライアントを設置した時の参加処理時間を表すモデル式は図6,表3,表4(センターから直接共有データを送信するため $\alpha(n)=1$)より、

$$T=4R+D(n)/Ps+D(n)/(B-E(n)) + Dq(n)/(Pc-E(n)) \quad (3)$$

となる。

5.3. イベントの圧縮方法

イベントの圧縮は、連続した同種類のイベントに対して行なう。

その圧縮方法は同じ種類のイベントが2つ以上連続して送信されたときに始めのイベントに差分情報だけを付加して送信することでイベントの数とデータ量を減らす方法である(図7)。

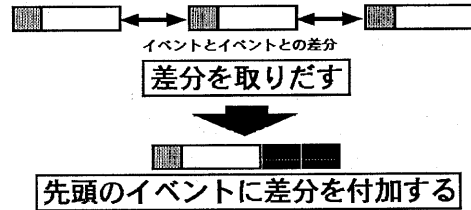


図7 イベントの圧縮例

この圧縮方式の場合、圧縮可能なイベントは、「同じ種類のイベント」「2つ以上の連続したイベント」「差分のとれるイベント」の条件を満たしたものである必要がある。

5.3.1. イベントの圧縮効果

既存のリアルタイムネットワークゲームでのログデータを基にイベントの内容を調べると、移動情報を持ったイベントが全イベントの53%あることがわかった。そこで移動情報イベントだけを選び、その差分を取り出して圧縮する方法を考えた。移動情報イベントは21バイトであり、付加する差分データは、参加者のID、上下、左右の合わせて3バイトのデータ量で済むことがわかった(図9)。

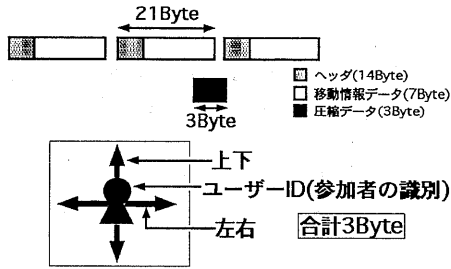


図9 イベント圧縮方法

また移動情報イベントの内87%が2つ以上連続したものであり、ログデータの解析よりそれが圧縮されるとイベントの数は11%に削減されることがわかった。つまり全体でイベントの数は65%に圧縮されることがわかる。また、イベントの平均データ量はログデータの解析より25%増加するので全体のイベント伝送量は81%に圧縮される(図10)。

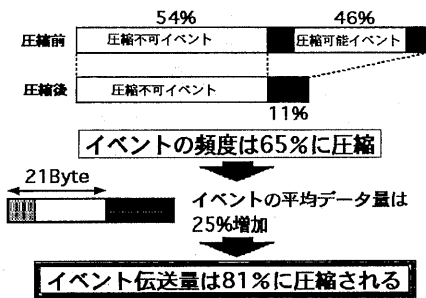


図10 イベントの圧縮率

6. 評価

評価は表5を基にして通信路帯域の狭い条件下で継続方式における疑似クライアントがある場合とない場合の途中参加処理時間の理論値の比較をおこなった。「既存参加者200人、参加希望者1人、通信路伝送容量128kbpsの条件の元で途中参加処理時間を求めると疑似クライアントのない場合では48secかかったが、疑似クライアントのある場合は1.6secしかかからなかった。疑似クライアントのない場合に比べて途中参加処理時間が約95%縮まることがわかった(図11)。

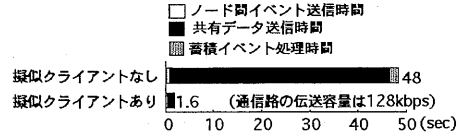


図11 通信路帯域が狭い場合の途中参加処理時間

7. まとめ

途中参加を行なうためのモデルとして中断方式と継続方式を提案し、途中参加処理時間の理論値を表すことでその問題点を示した。中断方式ではリアルタイム性が求められるアプリケーションでは使用できない可能性があり、また継続方式では通信帯域が狭い場合、共有データの送信で時間がかかる恐れがあることがわかった。そこでセンター内に疑似クライアントを設置しイベント圧縮を行なうことで、対象モデルの途中参加処理時間の理論値を短縮させることができた。このモデルでは、ある条件下の途中参加時間の理論値を95%削減することができた。

今後は疑似クライアントを実装させることで、方式の検証、適用領域について更に研究を進めていきたい。

参考文献

- [1] 石井裕: CSCW とグループウェア, オーム社(1994).
- [2] Alluisi, E.A: The Development of Technology for Collective Training :SIMNET a Case History, Human Factors, Vol.33, No.3, pp.346-362 (1991).