

P2Pに基づく分散コンピューティングの一実験(第二報)

— P2Pによるファイルおよびサービスバックアップシステムの開発と運用 —

市川 幸宏[†] 大東 俊博[†] 毛利 公美[†] 森井 昌克[†]

[†] 徳島大学 工学部 知能情報工学科 〒 770-8506 徳島市南常三島町 2-1
E-mail: †{ichikawa,ohigashi,mmohri,morii}@is.tokushima-u.ac.jp

あらまし P2P 環境での分散コンピューティングではネットワークの遮断や個々のコンピュータの事故などの原因によって、途中の計算結果が破棄された場合に再計算を行うための情報を管理する手法が重要となる。本稿では P2P クライアントが逐次計算するデータを部分共有化しながら、ネットワークの遮断やコンピュータの事故に強い分散コンピューティングを実現する方法を与える。本稿で提案する方式では Peer 全体でデータベースを分散して保持しているため、各 Peer が持っているデータを一定数収集することで従来サーバが保持していたデータベースを復元することが可能となる。すなわち、本方式は P2P によってファイルおよびサービスのバックアップを可能とする方式と考えられる。

キーワード P2P, 分散コンピューティング, グリッドコンピューティング, インターネット

A Distributed Computing Based on P2P Client(PartII)

-Development of File and Service Backup System by P2P-

Sachihiro ICHIKAWA[†], Toshihiro OHIGASHI[†], Masami MOHRI[†], and Masakatu MORII[†]

[†] Dept. of Information Science and Intelligent System, The University of Tokushima
Minamijosanjima-cho 2-1, Tokushima, 770-8506 Japan
E-mail: †{ichikawa,ohigashi,mmohri,morii}@is.tokushima-u.ac.jp

Abstract Recently, the distributed computing using personal computer is interested. The distributed computing requires a method to recovery the network troubles. The autonomous decentralized system has resistance for network troubles. However, generally projects using distributed computing are not the autonomous decentralized system but the client-server system. In this paper, we propose a new system using distributed computing that applies Peer-to-Peer. The system is autonomous decentralized system.

Key words P2P, distributed computing, grid computing, Internet

1. ま え が き

分散コンピューティングとは膨大なタスク(仕事)を小さく分割した上で多くのコンピュータで分散して処理をする方法である。分散コンピューティングはタンパク質の構造解析や暗号の強度評価などの膨大な計算量が必要なタスクへの利用で注目されている。具体的な実用例は、暗号解読を主として行う distributed.net [1] や地球外からの電波を解析する SETI@home [2], メルセンヌ素数以外で 100 万桁以上の素数の発見を目指す“素数探索プロジェクト” [3] などがある。これらに共通して言えることはタスクの分散化という意味では分散コンピューティングに属しているが、クライアント・サーバの接続形態を取っているためネットワーク

資源や計算機資源の分散化による特質を十分に活かしているとは言えないということである。

ネットワークや計算機資源の分散化を実現する技術として P2P と呼ばれる通信技術がある。著者らは文献 [4] において、P2P の特質を活かした分散コンピューティングの方式を提案している。文献 [4] では著者らの研究グループが運営している素数探索プロジェクトを対象に P2P 技術を導入する手法を与えている。具体的には P2P 技術によりネットワークや計算機資源を分散し、タスクの配布(素数候補の配布)やタスクの移譲(素数候補の管理と別クライアントへの移譲)というサーバ機能の分散化を実現した。サーバ機能の分散化によって、ネットワーク遮断やコンピュータの事故などに耐性がある信頼性の高い分散コンピュー

ティングを実現している。更に P2P 技術でタスクの計算の途中経過を共有することによって、タスク移譲時の探索の効率化を実現している。

本稿ではタスクやタスクの実行結果を P2P で部分共有化することで従来サーバに備わっていたデータベースを全て分散する方法を与える。データベースの部分共有化による利点は、記憶媒体の障害や不正アクセスなどでタスクおよびタスクの実行結果を保存しているデータベースが破壊された場合でも各 Peer から分散されたデータを回収することで元のデータベースを復元できることである。また本稿では P2P 通信で重要となる他 Peer の IP アドレスの共有方法を陽に与える。本稿で提案する方式はサーバ機能の分散化とデータベースの分散化を実現することで、ファイルおよびサービスのバックアップを可能にした分散コンピューティングの方式と考えられる。

2. 分散コンピューティング

分散コンピューティングは膨大なタスクを分割し、複数のコンピュータで処理する方式である。従来の分散コンピューティングはクライアント-サーバモデルであり、サーバがタスクを管理してクライアントの結果を受け取る。

2.1 素数探索プロジェクト

分散コンピューティングの一例として、著者らが運営している素数探索プロジェクトを紹介する。素数探索プロジェクトは分散コンピューティングの手法により、メルセンヌ素数以外で 100 万桁以上の巨大な素数を発見することを目的としたプロジェクトである。素数探索プロジェクトではサーバが各クライアントに素数の候補を配付し、クライアントが送信した素数判定結果を受信する。

素数探索プロジェクトのシステムは素数探索ソフト、サーバプログラム、以下に示す 4 種類のデータベース (以下 DB) から構成されている。

- 登録者 DB: 参加者の登録名とパスワード、参加者のプロジェクトへの貢献度を記録する
- 担当 DB: 各素数探索ソフトに移譲した素数候補を管理する。その素数候補の探索進行状況と素数のパラメータ、素数探索ソフトの途中経過報告時間を記録する。
- 素数候補 DB: 探索する素数候補を管理する
- 素数発見 DB: あるクライアント素数探索ソフトが素数を発見した場合、そのクライアントの登録者名と発見された素数、および素数の発見時刻を記録する。

素数探索プロジェクトでは上記の構成要素を用いて以下の手順で探索を行う。

(1) 素数候補要求

素数探索を行うクライアント (以下、クライアント) はサーバに対して素数候補を要求する。サーバはクライアントの認証後に素数候補 DB の素数候補を 1 つクライアントに送

信する。このとき担当者 DB に登録名と素数候補、および素数候補を送信した時刻を登録する。

(2) 途中経過報告

クライアントから途中経過報告が送信された場合、担当 DB の探索進行状況と報告時刻を更新する。また前回報告時から探索が進んだ割合のみ登録者 DB の探索数を加算する。

(3) 結果報告

クライアントにおける一つの素数候補の探索が完了、すなわち判定が終了すると素数か否かの結果をサーバに送信する。サーバは素数候補 DB から素数候補を 1 つクライアントに送信する。担当 DB の素数候補パラメータと報告時刻を更新し、前回報告時から探索が進んだ割合だけ登録者 DB の探索数を加算する。

探索結果が素数であった場合、その素数と探索担当者の登録名、および発見時刻を素数発見 DB に追加する。

素数探索プロジェクトではプロジェクトから外れたクライアントを把握するために、探索経過報告の期限を設けている。担当 DB ではクライアントから途中経過報告があった最終時刻を記録し、サーバ側で 1 時間毎に担当 DB の更新時刻の項目を調査する。あるクライアントの最終更新時刻が 2 週間以上経過している場合報告期限切れとし、そのクライアントはプロジェクトから外れたとみなす。報告期限切れ時には素数候補パラメータを素数候補 DB に戻し、他のクライアントからの候補要求時に再度移譲する。これにより、欠損なく全ての素数候補を探索することが可能となる。

2.2 分散コンピューティングにおける問題点

一般に分散コンピューティングではサーバは長期間接続がないクライアントをプロジェクト放棄とみなして、他のクライアントに放棄したタスクを依頼する。この場合、計算途中のタスクを新規のタスクとして扱うと、現在までに処理した計算結果が無駄になる (問題点 1)。また、ネットワークの遮断等によりサーバとクライアント間の通信が断たれた場合、通信切断中にサーバから新しいタスクを得ることができないため処理を中断せざるをえなくなる (問題点 2)。さらに、サーバのデータベースが記憶媒体の障害などによりサーバのデータを破壊された場合、プロジェクトの探索結果が消去されるという問題もある (問題点 3)。素数探索プロジェクトにおいても同様の問題点が生じる。これらの問題点はクライアント-サーバモデルの形態を取るゆえに起こることであり、P2P などの自律分散的なモデルによって解決が可能となる。

2.3 P2P

P2P (Peer-To-Peer) と呼ばれるクライアント同士の対等な接続形態を用いた自律分散システムが注目されている。P2P は従来のクライアント-サーバの接続形態とは異なりサーバを介さないため、耐障害性に優れ、クライアント同士でのファイルの共有が可能である等の様々な特徴を持つ。P2P に基づいた分散コンピューティングは、クライアントとサーバの接続だけではなく、クライアント同士でも接続し、グリッドコンピューティング

と呼ばれる形態となる。

3. P2P に基づく分散コンピューティング

本章では 2.2 節で挙げた 3 つの問題を解決する提案手法を与える。提案手法では P2P を利用して問題点を解決する。

問題点 1 に対しては、1 つのクライアントの探索途中結果をいくつかの他のクライアントにバックアップすることで解決する。これよりプロジェクトに参加しているクライアントは他のクライアントの探索途中結果を所有するので、あるクライアントが探索を途中で放棄し、他のクライアントがサーバからこの放棄した探索候補の探索許可を得ると、探索を途中から開始できる。

問題点 2 に対しては、クライアントは未探索の素数候補を所有するクライアントから未探索の素数候補を取得することで解決する。

問題点 3 に対しては、複数のクライアント間で素数候補の探索結果をバックアップし合うことで解決する。また、P2P を利用してクライアント同士が接続するためには IP アドレスを共有する必要がある。提案手法では、クライアントがサーバに初期接続するとき他のクライアントの IP アドレスを取得することで解決する。以下では上で述べた提案手法を具体的に説明する。

3.1 P2P による探索途中結果の共有

本節では各クライアント同士が P2P を利用して探索途中結果を共有し、探索放棄時に他のクライアントが共有している探索途中結果を用いて探索を再開する方法を説明する。本節で説明する方法によって、探索途中で放棄された素数候補を再計算することなく効率的に提案することができる。

素数探索プロジェクトでは素数探索ソフトとして Proth.exe [4] を利用している。Proth.exe はプログラムを停止しても途中から探索を再開できるように探索の途中経過をファイルとして保存することができる。この探索途中ファイルを他のクライアントにバックアップすることで、他のクライアントは探索放棄された素数候補を途中から探索できる。他のクライアントへのバックアップでは、探索途中ファイルを他のクライアントに送信する場合サーバを介して送信できるが、探索途中ファイルは約 1MByte の容量があり、複数のクライアントのファイル転送によって、サーバへの負担が増大する。そこで、クライアント間で直接ファイルの送受信を行う P2P 方式で探索途中ファイルをバックアップしサーバへの負担を軽減する。

図 1 に探索途中ファイルのバックアップと探索の再委託の流れを示す。

3.1.1 探索途中ファイルのバックアップ

素数探索ソフトが素数判定を行っている間、探索途中ファイルは計算の進行状況によって更新される。探索の再委託時に無駄な計算を最小限に抑えるためには常に最新の探索途中ファイルをバックアップする必要がある。

クライアントは 2 時間ごとに選択した他のクライアントに対し、自らが計算中の探索途中ファイルを P2P で送信する。探索途中ファイルを送信するクライアントを多くすることで探索を再委託可能なクライアントを増やすことができるが、探索途中ファイルを受信するクライアントを増やしすぎると、クライ

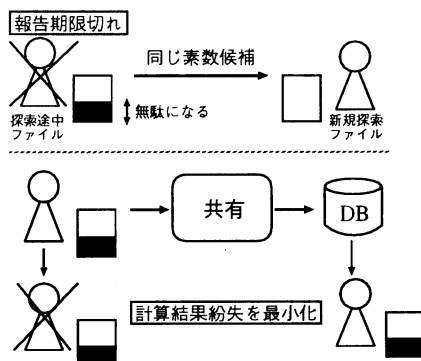


図 1 探索途中ファイルのバックアップと探索の再委託

アントのコンピュータの HD の容量を圧迫することになる。そこで、探索途中ファイルの送信先クライアント数および受信元クライアント数をそれぞれ 5 に制限する。さらにクライアントはサーバに探索途中経過を報告するときバックアップしている他のクライアントからの探索途中経過を含めて報告する。

3.1.2 探索途中ファイルのバックアップデータを利用した探索再開

クライアントは素数候補の探索を完了し結果をサーバに送信するとき、バックアップとして保持している探索途中ファイルの送信元クライアントが報告期限切れであるか否かをサーバに確認する。送信元クライアントが報告期限切れである場合探索途中ファイルから探索を再開する。報告期限切れでない場合は送信元クライアントが現在も素数探索中であるとみなし、未探索の素数候補を探索する。

3.2 P2P による素数候補の共有

本節では P2P を利用してクライアント同士が素数候補を共有する手法を説明する。この手法を用いることでサーバとの接続が断たれている際に探索を続行できないという問題を解決することができる。

ネットワーク障害で他のクライアントと P2P 通信は可能であるがサーバとの通信が不能という状況が起こりうる。このときクライアントは素数探索を完了しても探索結果をサーバに送信できない。クライアントが素数候補を複数保持していればサーバとの通信が回復するまで他の素数候補を探索し続けることができる。しかしながらクライアントが保持している全素数候補の探索が終了してもなおサーバに探索結果を送信できなければ、探索を続けることが不可能になるため、時間の無駄が生じる。各クライアントが動作しているコンピュータには性能差があるため、一方のクライアントが保持している全素数候補の探索が終了しても、もう一方のクライアントは未探索の素数候補をいくつか保持しているという状況が起こりうる。その 2 つのクライアントが P2P 通信可能な場合、未探索の素数候補を保持するクライアントから全素数候補の探索を終了したクライアントへ素数候補を移譲することで、クライアントが待機状態になることを防げる。

素数探索プロジェクトではクライアントが一度にサーバから

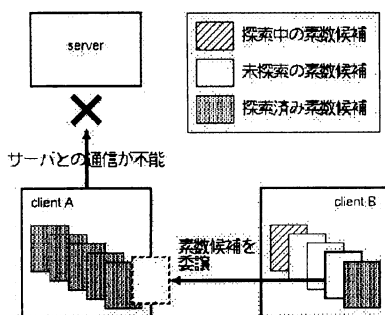


図 2 素数候補の移譲

受信する素数候補を 1 個にしている。素数候補の移譲を実現するためにクライアントが一度に保持できる素数候補の数を 5 つに増やし、クライアント同士が P2P を利用して素数候補を共有可能にする。図 2 にクライアント間での素数候補移譲の流れを示す。

3.2.1 サーバ・クライアント間の素数候補の移譲

サーバへの探索結果を報告の際にサーバと通信することが不可能な場合、クライアント側で探索結果を保持し未探索の素数候補を探索する。その後サーバとの通信が回復したときに探索結果を送信し、送信した個数の素数候補をサーバから受信する。

3.2.2 クライアント間の素数候補の移譲

サーバへ探索結果を送信できず保持している素数候補探索がし終えたとき、P2P 通信可能な他のクライアントが未探索の素数候補を保持していればそのクライアントから素数候補を受信する。また、素数候補を移譲したクライアントはサーバと接続したとき素数候補を移譲した数の素数候補をサーバから受信する。

3.3 クライアント間の探索結果のバックアップ

サーバに深刻なダメージが発生した場合サーバ上に存在する DB を復旧する必要がある。本節では P2P を利用しクライアント間で探索結果をバックアップする方法を説明する。この方法を用いることで各クライアントのバックアップから容易にサーバの DB を復旧できる。

素数探索プロジェクトではクライアントに配布した素数候補が素数であるか否かの探索結果はサーバ上での DB で管理している。仮にこの DB が失われれば現時点までの素数判定結果が全て失われてしまう。

探索結果を共有するためにクライアントはサーバに探索結果を送信する際に、他のクライアントとの通信が可能であれば P2P を利用し探索結果をクライアントに送信する。これによって、各クライアントは他のクライアントの探索結果をバックアップする。サーバに深刻なダメージが発生し探索結果の DB が破損した場合、各クライアントが分散バックアップしているため探索結果の DB の復旧が可能である。

3.4 IP アドレスの共有

P2P 通信を実現するためには IP アドレスの共有が必要である。3.1, 3.2, 3.3 節で説明した探索途中結果の共有、素数候補の共有、探索結果のバックアップを P2P 通信で行うために必要となる IP アドレスの共有方法について説明する。

クライアントはサーバに初期接続する際に登録者 DB に登録されるクライアントの IP アドレスをランダムに 25 個取得する。クライアントは 25 個の IP アドレスのうち主に 5 つの IP アドレスを利用して P2P 通信を行う。クライアントは P2P 通信を行う際にまずサーバから取得した IP アドレスのクライアントが接続可能か不可能かをチェックする。クライアントは接続に成功すれば接続した時間を記録する。接続不可能であり、かつ接続した時間が一定期間過ぎていれば、クライアントはその IP アドレスが利用不可能になったと判断しその IP アドレスを削除する。クライアントは保持している IP アドレスの数を常に監視し、その数が一定の数より少なくなると P2P 通信可能なクライアントに IP アドレスを要求し、自分自身が保持する IP アドレスを増やす。

4. P2P に基づく分散コンピューティングの実装

本章では提案手法を実際に実現したシステムについて述べる。サーバは Java、クライアントはインターフェイスを VC++、通信部分を Java で作成した。DB は PostgreSQL を使用した。

4.1 概要

本節では本システムの動作概要について述べる。

Step1. 素数探索ソフトを起動し、Socket クラスを利用し、

```
Socket servsock = new Socket(port); (1)
```

ポートを指定してサーバと接続する。現在ポート番号は 9700 である。

Step2. サーバは素数候補を DB から取得する。データを送信するためには、(1) のようにソケットを作成し、ポートを指定して DataOutputStream クラスの out を

```
DataOutputStream out (2)
```

```
out.writeUTF(data); (3)
```

でクライアントに送り、素数探索が開始される。

Step3. クライアントは定期的に探索途中経過をサーバに送信する。

```
Socket cliensock = new Socket(port); (4)
```

(3) で文字列 data をサーバに送信する。

Step4. クライアントの素数探索が終了すると、Proth.exe の出力ファイル Proth.log が出力される。Proth.log に composite か factor の文字列が記録されていれば、探索結果は素数の可能性はないと判断する。maybe か may, prime! なら探索結果は素数の可能性があるかと判断する。探索結果をサーバに送信する。

(4),(3) を用いてサーバに送信する。

Step5. サーバから新たな素数候補を受け取り、探索を続ける。

(1) でソケットを作成する。

```
DataInputStream in (5)
in.readUTF(); (6)
```

サーバからストリームで送信されたデータを readUTF で読む。読み取った値は文字列として readData に入る。

以降クライアントソフトを停止させるまで Step3~Step5 を繰り返す。

4.2 システムの通信部

本システムは素数候補を送信し結果をデータベースで管理するサーバ、素数候補を受信し素数を探索するクライアントで構成される。クライアント-サーバ間の通信を“クライアント-サーバ通信部”と定義する。クライアント間の通信を“P2P 通信部”と定義する。

4.2.1 クライアント-サーバ通信部

● 探索途中経過ファイルを送信

受信した他のクライアントのバックアップファイルは receive.dta にファイル名とソフトウェアを区別するための ID(以下 MID) を書き込む。(4) でソケット作成,(6) で受信したデータを読む。

```
File file = new File(バックアップファイル);
file.write(readData);
```

他クライアントのバックアップファイルが存在する場合、探索途中経過報告をサーバに送信する際に他クライアントのバックアップファイルの探索状況も同時に送信する。(4) でソケット作成。

```
out.writeUTF(バックアップファイル + 探索途中経過報告);
```

Step1. 探索途中経過送信後、他クライアントのバックアップファイルが存在するか確認する。

```
File fp = new File(recieve.dta);
fp.readline();
```

Step2. 他クライアントのバックアップファイルが存在する場合、

```
File file = new File(バックアップファイル);
file.exist() == true;
```

ファイル名 (Proth_[MID].dta) からコピー元のクライアント、ファイルの内容から素数候補パラメータ及び探索状況を得る。

```
File file = new File(Proth_[MID].dta)
data = file.readLine();
```

これらの情報をバックアップファイル探索途中経過報告コマンドで送信する。(3) でデータを送信。

Step3. バックアップファイル探索途中経過報告コマンドを受信したサーバは担当 DB 内の対応する素数候補の探索状況の値を更新する。報告時刻は更新しない。受信したデータを (6) でファイルを読む。PostgreSQL を利

用して、データを更新する。

```
sql="UPDATE DB の名前 SET mid="+MID+",
name="+登録名+" where oid="+素数候補;
statement.executeUpdate(sql);
```

● 探索結果を送信

一つの素数候補の探索が終了した場合、素数の判定が探索結果ファイルに出力される。サーバに探索結果を送信する。クライアントの素数探索が終了すると、Proth.log が出力される。Proth.log を読み解析する。

composite か factor の文字列が記録されていれば、探索結果は素数の可能性はないと判断する。maybe か may, prime! なら探索結果は素数の可能性がある判断する。(4) でソケット作成し、(3) で探索結果を送信する。

Step1. 探索結果をサーバに送信する。

```
File file = new File(Proth.log)Proth.log
indexOf(composite or facotr or maybe or may or prime!);
```

(4) でソケットを作成し、(3) でサーバに送信する。

Step2. サーバは受信したデータを DB に保存する。

Step3. クライアントに素数候補を渡す。

● 期限切れの問い合わせ

クライアントは素数探索終了後に、探索結果をサーバに送信する。他クライアントのバックアップファイルが存在する場合、バックアップ元のクライアントが期限切れであるか否かをサーバに確認する。

Step1. 結果報告コマンドにバックアップファイルの前の所得者の MID を添えてサーバに送信する。(4) でソケット作成し、writeUTF で MID 送信する。

Step2. サーバは結果報告コマンドを受信すると、MID に対応する担当 DB 内のクライアントが探索期限切れか確認する。期限切れでなければサーバは新たな素数候補をクライアントに送信する。Step5 へ。

```
(1) でソケットを作成し、
素数候補を小さい順に DB から取り出す
sql="select * from ticket2 order by form.k,n";
data=statement.executeQuery(sql);
(3) で送信する。
```

Step3. 期限切れならば、担当 DB の担当クライアントを結果報告コマンドを送信したクライアントに変更し、探索引き継ぎ許可コマンドをクライアントに返信する。

```
DataOutputStream out (7)
```

```
out.writeUTF(引継ぎ許可コマンド); (8)
```

Step4. 再委託許可コマンドを受信したクライアントはバックアップファイル名を Proth_[MID].dta から Proth.dta へ変更し、Proth.dta から素数候補パラメータを読み取る。その後クライアントが保持する素数候補リストに引き継いだ素数候補を追加する。

Step5. 再委託した素数候補あるいは持っている未探索の素数候補で探索を再開する。

4.2.2 P2P 通信部

● 探索途中経過ファイルのバックアップ

クライアントが起動時に 30 分毎に途中経過ファイルを他クライアントへ転送する。以下のプログラムは 30 分毎にスレッドを走らせている。

```
if(tmptime[0] == 0L)
tmptime[0] = System.currentTimeMillis();
else
tmptime[1] = System.currentTimeMillis();
ttime = tmptime[1] - tmptime[0];
...
if(mintime >= lmin) new Thread(sendp2).start();
```

Step1. ファイル転送先クライアントを設定する。初回起動では IP テーブルから任意のクライアントを選択し、選択したクライアントにバックアップ転送許可願コマンド及び途中経過ファイルの更新時刻を送信する。

Step2. バックアップ転送許可願コマンドを受信したクライアントは、他クライアントのバックアップファイルを既に 5 つ保有しているか、前回転送したファイルと今回受信予定のファイルの更新時刻が等しいか否かを調べる。いずれかの条件を満たす場合は Step5 へ。

Step3. 送信許可コマンドを相手クライアントに送信する。コマンドを受け取ったクライアントは、素数探索ファイル (Proth.dta, Proth.dtf) を相手クライアントに送信する。IP テーブル中の、転送に成功したクライアントは、次の転送動作時に同じ転送先クライアントを選択する。

Step4. ファイルを受信したクライアントは、相手クライアントのマシン ID (MID) をファイル名に付加し (Proth-[MID].dta) 保存する。以上で転送動作を終了する。

Step5. ファイルの受信を拒否するコマンドを相手クライアントに送信する。相手クライアントはコマンド受理した後転送動作を終了する。

● 探索結果のバックアップ

素数候補を探索終了時に探索結果が出力される。その探索結果を他のクライアントに分散バックアップする。

Step1. IP テーブルから任意のクライアントを選択し、探索結果とユーザの ID を送信する。

```
File file = new File(IP テーブル);
IP adress = file.readLine();
writeUTF(探索結果, ユーザの ID);
```

Step2. 受信したクライアントは特定のフォルダに保存する。

● IP テーブルの共有

サーバから、初期接続時に複数 IP アドレスを取得する。その後、設定された閾値と現在の IP アドレスの数を比較し、閾値よりも IP アドレスの数が少なければ、他のクライアントと IP テーブルを共有し、IP アドレスの数を増やす。

Step1. 設定した閾値と現在の IP テーブルの IP アドレスの数を比較する。

```
while(IP table.readLine)
IP_count++;
if(IP_count < ini)
else
```

Step2. 比較した結果、IP アドレスの数が少ない場合のみ、他のクライアントと IP テーブルを共有する。

writeUTF(IP テーブル); (4) でソケット作成し、(6) で IP テーブルを読み、IP アドレスのパターンマッチングを行い、取得していない IP アドレスを自分の IP テーブルに書く。

● 素数候補の受け渡し

クライアントが保持している全素数候補の探索が終了し、かつサーバとの通信に失敗した場合は素数候補要求動作を実行する。

Step1. IP テーブルから任意のクライアントを選択し、素数候補要求コマンドを送信する。

```
writeUTF(素数候補要求コマンド);
```

Step2. コマンドを受信したクライアントは、自らが保持している未探索の素数候補が 2 個以上存在する場合、素数候補受渡しコマンドと素数候補パラメータを一組送信する。

Step3. 素数候補を受信したならば、クライアントが有している素数候補リストにパラメータを追加し、その素数候補の探索を行う。

5. む す び

本稿ではクライアントに P2P 技術を用いる分散コンピューティングの方式を提案した。本手法は P2P 技術の特徴を活かし全ての資源を有効に活用することで効率的に処理を行うことが可能である。さらにネットワークが遮断されサーバと接続が不可能になったとしても、P2P の技術を用いることで計算結果紛失を可能な限り最小化することができる。本稿で提案した方式は計算が効率的でかつ計算結果の紛失などに対して耐性があるため、ネットワークの事故などが存在する環境において優れた分散コンピューティングの方式と考えられる。

文 献

- [1] "distributed.net," available at <http://www.distributed.net/>
- [2] "SETI@home," available at <http://setiathome.ssl.berkeley.edu/>
- [3] "素数探索プロジェクト," available at <http://n173.is.tokushima-u.ac.jp/>
- [4] 竹内崇洋, 畑島賢治, 市川幸宏, 大東俊博, 毛利公美, 森井昌克, "P2P に基づく分散コンピューティングの一実験 -素数探索プロジェクト-", 信学技報 OIS2004-4, pp.19-24, May 2004.