

IMAP エージェントによる重要メール緊急通知システム

陳 修遠^{1,a)} 村上 登志男^{2,b)} 山井 成良^{1,c)} 中川 令^{1,d)}

概要：電子メールはインターネットにおける社会活動を支える最も重要なコミュニケーションツールである。しかしリアルタイム性に乏しく、たとえば深夜に非常に緊急で重要なメールが届いたとき、受信者が気付かない可能性がある。IMAP (Internet Message Access Protocol) とは、インターネットなどの IP ネットワークで標準的に用いられる、電子メール (e メール) を受信するための通信規約 (プロトコル) の一つである。利用者が自分宛での電子メールを保管しているメール受信サーバにアクセスし、新着メッセージを確認したり一覧から必要なものを選んで手元に受信する手順を定めている。ユーザにとって重要なメールを見逃さないために、IMAP により電子メールサーバに常時接続して新着メッセージの監視を行い、ユーザが設定した条件を満たした緊急性の高い新着メッセージを別の手段でユーザに直ちに通知するシステムを提案する。

Emergency Notification System for Important Emails Using IMAP Agent

1. はじめに

現在、インターネットは世界中で広く使われており、インターネットは生活に必要不可欠なものとなっている。総務省の令和3年情報通信白書 [1] からみると、2020年のインターネット利用率 (個人) は83.4%となっている。そしてインターネットの利用目的は、「電子メールの送受信」が最も多い。電子メールはインターネットにおける社会活動を支える最も重要なコミュニケーションツールであるが、リアルタイム性に乏しく、たとえば深夜に非常に緊急で重要なメールが届いたとき、受信者が気付かない可能性がある。さらにそれだけではなく、メールソフトのプッシュ通知機能が使えなくなることがあり、メールが届いてもユーザに通知されないことがある。ユーザにとって重要なメールを見逃さないために、IMAP (Internet Message Access Protocol) により電子メールサーバに常時接続して新着メッセージの監視を行い、ユーザが設定した条件を満

たした緊急性の高い新着メッセージを、電子メール以外の手段でユーザに直ちに通知するシステムを提案する。

さまざまなメールソフトの製品があるが、本研究には普段使っているメールソフトである Gmail, Yahoo!メール, Outlook の三つを検討した。これらのメールソフトにはそれぞれのメールを重要と判断する機能が付属しているが、実際に使用したときの効果はどうだろうか。例えば Gmail では、さまざまな情報を利用して、重要マークを付けるメールを自動的に判断している。この判断機能はユーザに通知することができないため、ユーザにとって柔軟に使うことができない。本研究の目的のうちの一つは、ユーザにとって使いやすい重要なメールの判断システムを作ることである。

緊急通知の方法にはさまざまな方法が考えられるが、たとえば SMS (Short Message Service) を用いる方法を検討している。これは多くのスマートフォン (iOS と Android) では「おやすみモード」中であっても特定の発信者からのメッセージを受信した場合にユーザに通知する機能があるためである。また、HTTP 通信により SMS メッセージを送信すると音声電話の商用サービスが存在し、IMAP エージェントから利用することも可能である。なお、ユーザにとって柔軟な通知方法を選ぶため、なるべく多くの通知手段ができることを目指す。

¹ 東京農工大学
Tokyo University of Agriculture and Technology

² 学習院大学

Gakushuin University

a) Rockcxy@net.cs.tuat.ac.jp

b) toshio.murakami@gakushuin.ac.jp

c) nyamai@cc.tuat.ac.jp

d) fr3660@go.tuat.ac.jp

2. 背景

2.1 IMAP と IMAP エージェント

電子メール (e-mail) とは、パソコンや携帯電話、スマートフォンなどの情報機器同士が、専用のメールソフトを使って、インターネットなどのネットワークを利用して情報をやりとりする機能である。やりとりできる情報は文章 (テキスト) だけでなく、文書ファイルや画像などを添付ファイルとして扱うことができる。IMAP (Internet Message Access Protocol) は、電子メールサーバ上の電子メールにアクセスし操作するためのプロトコルである。クライアントとサーバが TCP を用いて通信する場合、通常サーバ側は IMAP4 ではポート番号 143 番、IMAP over SSL (IMAPS) では 993 番を利用する [3]。一方でユーザとユーザの間のメールの送信では、SMTP (Simple Mail Transfer Protocol) によって、電子メールサーバ間でのどのような作業が行われるかを規定され、電子メールを確実に効率的に送信することを可能としている。受信では、同様に POP3 (Post Office Protocol version 3) と IMAP の二種類が主に利用されている。両者ともに、電子メールサーバに保管されているメールを読むという基本的な機能は共通しているが、それぞれに特徴を持っている。IMAP4 は最も普及したバージョンであり、単に IMAP といった場合は IMAP4 を指すことが多い。

電子メールサービス事業者はメール受信時にメッセージの内容を検査し、検査結果に基づいて分離、隔離、削除、転送等の処理 (以下、検査・処理) を行う製品やプログラムを導入している。しかし、電子メールサービス事業者が提供する検査・処理機能では不十分であったり、ユーザに応じたカスタマイズが十分行えなかったりする場合も多い。このような場合、電子メールサーバに導入された上記の製品やプログラムにユーザが新たな検査・処理機能を追加することは通常は困難であるため、ユーザはたとえば MUA (Mail User Agent) にプラグインプログラムを導入して検査・処理を行うことになる。特に IMAP サーバや Web メールのように電子メールサーバ上でメッセージ管理する形態では、MUA ではメッセージを実際に取得するまでメッセージの内容を検査することができず、検査・処理が遅れることが問題となる。例えば、重要な人がユーザーに重要なメールを送った場合、ユーザは最初の時間にメッセージを見ることができないか、メールが来たことに気づかない。また、特に Web メールでは、ヘッダ情報など必要な情報が端末で取得できるとは限らないため、端末側でメッセージの内容に基づく検査・処理を行うことは困難である。それに対して、筆者は電子メールサーバに常時接続する IMAP エージェントによりこれらの問題を解決する方法を提案した。IMAP エージェントは IMAP プロトコルに基づいた電子メールサーバとの通信を行い、サーバ上の

メールや電子メールサーバに関する各処理を実施するシステムである。IMAP エージェントは処理の自由度が高く、多目的に応用可能な柔軟性を持つことが特長の一つである。また、多くの Web メールサービスでは同時に IMAP サーバも利用可能であることから、Webメールの利用者に対しても検査・処理機能を追加することが可能である [2]。

2.2 既存の重要メール判別方法の問題点

現在既存電子メールサービス事業者 (Gmail[4], Outlook[6], Yahoo!メール [5]) の判別標準を参考する。

Gmail Gmail ではさまざまな情報を使用して、メールに重要マークを付けるかどうかを自動的に判断する。Gmail が重要と判断したメールの横には、黄色の重要マーク Importance marker が表示される。重要マークが付いていないメールには、色が無い形だけのマークが表示される。重要マークが付いたすべてのメールを確認するには、Gmail で検索キーワードに「is:important」を指定して検索する。

Outlook Outlook ではメッセージの重要度を高または低に設定する。メッセージを必ず読んで欲しい場合は、メッセージの重要度を高く設定する。メッセージが単なる FYI の場合、または仕事以外の関連トピックに関するメールを同僚に送信するなどといった場合には、メッセージの重要度を低く設定できる。また、受信したメールの重要度を変更することもできる。

Yahoo!メール Yahoo!メールでは自動振り分けという機能がある。特定のメールアドレスから送られてくるメールを自動で振り分けるフィルターを作成して、このアドレスから送られてくるメールの中で、他の項目のフィルターを設定することで、「重要というキーワードが件名中にあるメールのみ振り分ける」といった設定や、「本文中に特定のキーワードを含まないメールのみ振り分け」など細かい設定を行うこともできる。

以上が電子メールサービス事業者提供されている重要なメールの判別方法だが、これでは不十分であり、使用中電子メールソフトがユーザの思い通りに機能しない。電子メールサービス事業者の長所と短所を整理する。Gmail と Outlook は重要マークと重要度を設定した上、自動的に判断する。しかし、すべてのユーザが同じ電子メールサービスソフトを使用しているわけではないため、たとえば Outlook ユーザが Gmail ユーザにメールを送信する際に、Gmail 重要マークの機能が使えないため、Gmail ユーザはこのメールが重要メールであることを認識していないかもしれない。Yahoo!メールは重要なメールの判別機能がない。その代わりに自動振り分け機能がある。メールをうまく分類することは可能だが、あくまで他のフォルダに転送するだけであり、重要なメールが届いていることをユーザーに知らせることはできない。

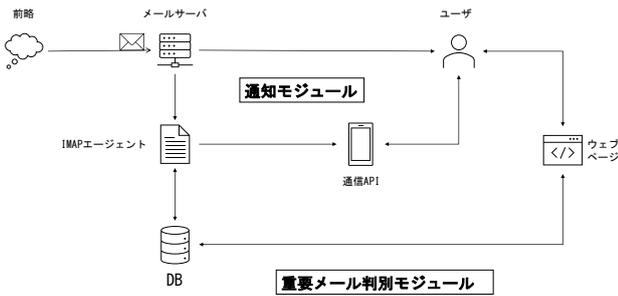


図 1 全体システムの構成

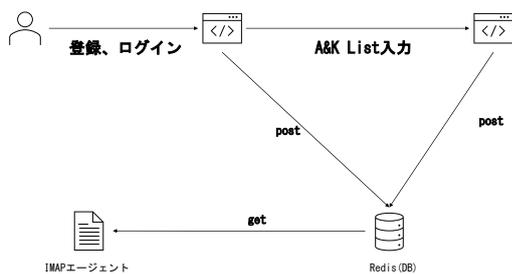


図 2 情報入力システムの仕組み



図 3 Web ページの表示例 (上)

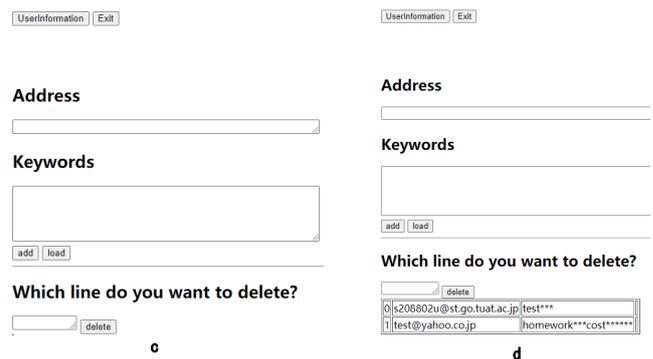


図 4 Web ページの表示例 (下)

3. システムの設計

システムの全体は、メールシステム、重要メール判別モジュールと通知モジュールの3つで構成されている。メールシステムは社会的に広く使われている。そして、主な部分は IMAP エージェントをめぐる重要メール判別モジュールと通知モジュールである。重要メール判別モジュールと通知モジュール部分の紹介は節 3.1 と節 3.2 で説明する。全体システムの構成を図 1 に示す。

3.1 重要メール判別モジュール

節 2.2 の問題点については、本研究で対処している。ユーザーに Address & Keywords List(以下 A & K List) を入力のため、MVC (Model-View-Controller)[7] で作る Web ページから個人ページに簡単に登録・ログインできる。MVC は、ユーザインターフェイス (ビュー)、データ (モデル)、およびアプリケーションロジック (コントローラー) を分離するために使用されるデザインパターンである。ユーザーに関するすべての個人情報は、IMAP エージェントコードの設定に隠されている。情報入力システムの仕組みを図 2 に示す。

流れは以下のプロセスと図 3 の a, b 及び図 4 の c, d で説明する。

a a は Login System であり、未登録ユーザーの登録も提供している。携帯電話番号またはメールアドレス、ユー

ザ名、パスワードを入力してログインする。その理由は、アカウント登録時に携帯電話やメールアドレスのいずれかが間違っていた場合、正しい方でログインできるようにするためである。登録情報を入力し、送信をクリックすると、ユーザーが正しい情報を入力したかどうかを確認するためのプロンプトボックスが表示される。

b b は個人情報変更画面であり、a で述べたように、アカウント登録時に誤った情報を入力した場合、正しい情報でログインした後に、この画面で変更することができる。

c c は My Page であり、ここでユーザーは自分の A & K List を作成することができる。追加、削除、導出機能も用意されている。

d d は c を基準とした現在の A & K List を示した。番号、メールアドレス、キーワード3つで構成されている。変更・削除したいリストを数字ですぐに見つけることができる。

ユーザーは Webpage で登録し、自分の A & K List を作る。Web ページは Login System と My Page 二つ機能がある。データ操作 (登録, ログイン, 変更, リスト入力, リスト削除など) は、データベース上で行われる。Web ページのデータは全てデータベースに post して、IMAP エージェントはデータベースにも接続できるので、データを get できる。これは、例えばユーザーがデバイスを変更したり、

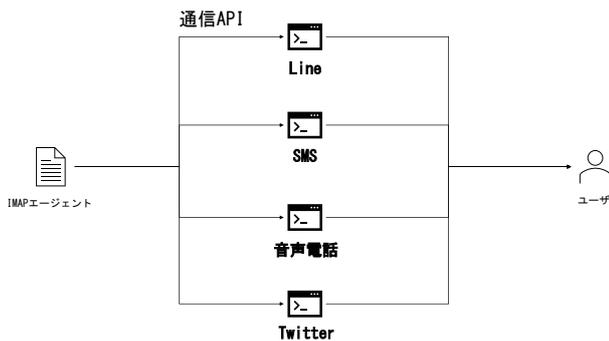


図 5 通知モジュールの構成



図 6 iOS（左）と Android（右）の動作確認

別のブラウザに変更したりした場合に、ローカルキャッシュがクリアされてもデータに影響を与えないようにするためである。ユーザの誤入力を防ぐため、誤入力防止機能つまり情報を修正する機能も搭載している。

3.2 通知モジュール

通知モジュールの機能は、IMAP エージェントと通信APIの接続を利用して、ユーザに情報を通知することである。メールソフトを使用していると、メールが来ても通知されないことがある。あるいは、スマートフォンがマナーモードやおやすみモードになっているため、メールアプリからの通知を受け取れない場合もある。しかし、いかなる場合であっても重要なメールを通知することは必要であり、本研究で取り上げるべきことのひとつである。そこで、通知方法や通知期間をいかに柔軟にユーザに設定させるかの解決方法を次に提案した。まず、ユーザが通知方法を選択する必要がある。この研究では、以下のような方法（SNS、音声電話、LINE、Twitter）を検討した。どちらを使うか指定するのではなく、ユーザに選んでもらうということである。通知モジュールの構成を図5に示す。

調査の結果、特定の時間に対して、特定の連絡先やソフトだけ通知音を鳴らすことができると判明した。現在最も使われているスマートフォンには、アップルのiPhoneとグーグルのAndroidの2種類がある。今回の研究では、この2種類のスマートフォンのOSに対して、それぞれ異なる対応付けがなされている。iPhoneでは、iOS 15に集中モード [8] が導入され、iPhone 7以降で利用できる。集中モードを設定する時、特定のアプリとユーザからの通知を許可することを提案している。集中モードで通知を許可されているユーザ以外からは、通知を受け取らないように設計した。仕事、パーソナルな時間、睡眠、フィットネス、マインドフルネス、ゲーム、読書、運転向けなどといった集中モードはさまざまであるため、時間や場所を設定することで、より柔軟に使用することができる。Androidを搭載したスマートフォンは数多く、ブランドもさまざまであるため、ここでは一つ簡単な例、Digital Wellbeing [9] という

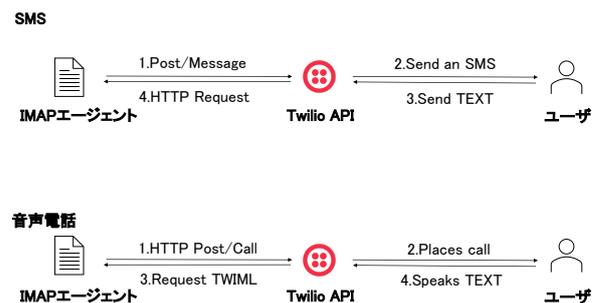


図 7 Twilio API（SMS、音声通話）の仕組み

アプリを紹介する。Digital Wellbeingの保護者による使用制限は、フォーカスモードをタップすることで、フォーカスモード設定画面を開くことができる。「フォーカスモード」のON/OFFを切り替えれば、特定のアプリからの通知管理でき、たとえば特定な時間帯にチャットのアプリからの通知をシャットダウンして使えないようにすることが可能である。Google Playストアから直接ダウンロードできるため、メーカーを問わず、Androidを搭載したスマートフォンであれば利用可能である。これは数あるGoogleアプリの中の一つに過ぎず、ユーザが自分の好きなアプリを使ってこの機能を実装することも可能である。今回の研究にはiPhone11 Promax（iOS側）とASUS Z01GD（Android側）を動作確認した。iOSとAndroidの動作確認を図6に示す。

ユーザが新着メッセージを知るには、SMSと音声通話が最も効果的な方法である。今回はTwilio社の製品を検討した。Twilio [10]は電話やSMS、ビデオ、チャット、SNSなど世の中にある様々なコミュニケーションチャンネルをWeb、モバイルアプリケーションとつなぐ「クラウドコミュニケーションAPI」であり、充実したサービスに加え、利用料金も安い。Twilio API（SMS、音声通話）の仕組みを図7に示す。

ユーザの選択肢を増やすため、SMSと音声電話以外、人気のあるSNS（LINEとTwitter）も検討した。特にTwitterは簡単に閲覧でき、多くの人が利用しているため、

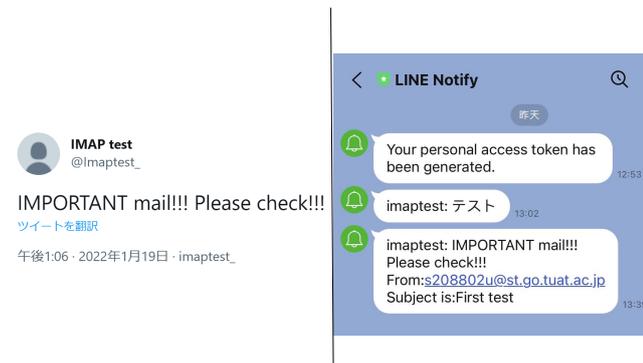


図 8 Twitter (左) と LINE (右) の表示例

次節のフェイルセーフ機能にも導入した。Twitter (ツイッター) [11] は SNS であり、「ツイート」と呼ばれる半角 280 文字 (日本語, 中国語, 韓国語は全角 140 文字) 以内のテキストや画像, 動画, URL を投稿できる。Twitter は Developer Platform という便利な機能がある。Dashboard で自分のプロジェクトを作成し, 管理することが可能である。Essential, Elevated, Managed の三つのアクセスレベルがある。いずれも無料であり, Twitter API v2 にアクセスすることができる。Elevated access の申し込みが必要だが, 現時点で申し込みが完了し利用を開始した状態となった。3つのアプリケーション環境 (開発環境, ステージング環境, 本番環境) と 200 万ツイート/月の取得ができる。LINE[12] は, 利用者が相互に本アプリケーションをインストールしておけば, 通信キャリアや端末を問わずに, 複数人のグループ通話を含む音声通話やチャットが可能である。LINE も Twitter と同じ開発者環境 LINE Developer コンソールがあるが, 動作確認だけのため, 代わりに本研究は LINE notify で実行した。Twitter と LINE の結果展示を図 8 に示す。

3.3 フェイルセーフ機能

フェイルセーフ [13] とは, 特定の種類の故障が発生した場合に, 他の機器や環境, 人への危害を最小限または全く発生させない方法で, 本質的に反応する設計上の特徴や習慣のことである。本研究では, 火災や停電などの災害により, ユーザの被害を最小限に抑えるために, フェイルセーフ機能を導入した。フェイルセーフ機能は IMAP エージェント多重化とデータベースのレプリケーション2つの部分から構成される。

IMAP エージェント多重化は IMAP エージェントプライマリと IMAP エージェントセカンダリで構成されている。IMAP エージェントセカンダリは電子メールサーバとデータベースを接続するだけでなく, IMAP エージェントプライマリも常時チェックを行う。ここで死活監視という手段を検討した。死活監視 [14] とは, 機器やシステム, ソフトウェアなどの対象が動作しているかどうか外部から

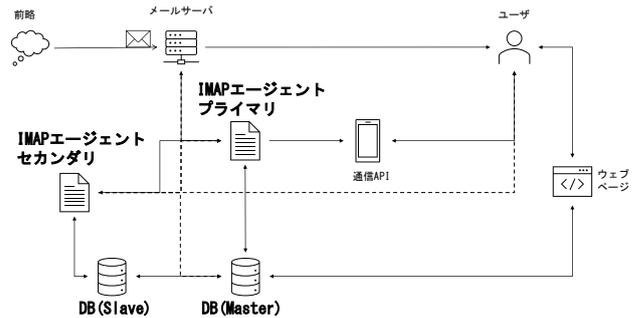


図 9 IMAP エージェント多重化の構造

定期的・継続的に調べることで, 特に専用の装置やソフトウェアなどによって自動的に調べ続けることである。一定時間応答がない場合は対象が機能を停止しているとみなし, 管理画面や電子メールなどを通じて管理者に通報・警告したり, 自動的に再起動させたりするようになっていることが多い。死活監視を活用することで, IMAP エージェントプライマリがある原因によってダウンしたときに, システム全体が正常に動くように, IMAP エージェントセカンダリは IMAP エージェントプライマリの機能をすべて継承する。IMAP エージェント多重化の構造を図 9 に示す。

次に, データベース多重化について述べる。Redis[15] は自身でマスタースレーブ型のレプリケーションによる冗長構成をとることができる。Redis は非同期レプリケーションを採用している。マスターは複数のスレーブを持つことができる。スレーブも読み込みクエリを受け付けることができるため, 負荷分散目的でも使用することができる。

step 1 Slave は Master との接続に成功した後, sync コマンドを送信する。

step 2 Master は, ディスク保存処理の開始コマンドを受信し, 受信したデータセットの修正コマンドをすべて収集した後, バックグラウンド処理を実行して完全な同期を完了した後に, データファイル全体を Slave に転送する。

step 3 フルレプリケーション。Slave は, データベースファイルのデータを受け取り, 保存し, メモリにロードする。

step 4 インクリメンタルレプリケーション。Master は, 新たに収集した修正コマンドをすべて Slave に順番に渡し続け, 同期を完了させる。なお, Master に再接続される時, フルシンク (フルレプリケーション) が自動的に実行される。

Master 1 台, Slave 2 台, スレーブは読み出しのみ可能で, 書き込みはできない。Slave が Master から切断された場合, 以前のマスター・スレーブ関係を確立するために, 新たな接続のスレーブが必要である。Master がダウンした後も, マスター・スレーブ関係も存在し, Master の再

```
UnseenMessage:
-----
! ! ADDRESS & KEYWORDS! !
--number is:
0
--Address is:
s208802u@st.go.tuat.ac.jp
--keywords is:
test

*****UID is:
238
*****From:
s208802u@st.go.tuat.ac.jp
*****Subject is:
First test
*****Body is:
--00000000000233f1005c9a7ec41
Content-Type: text/plain;
This is the first test.
```

図 10 動作確認

```
*****UID is:
238
*****From:
s208802u@st.go.tuat.ac.jp
*****Subject is:
First test
*****Body is:
--00000000000233f1005c9a7ec41
Content-Type: text/plain, charset="UTF-8"
This is the first test.
--00000000000233f1005c9a7ec41
Content-Type: text/html, charset="UTF-8"
<meta http-equiv="Content-Type" content="text/html"
--00000000000233f1005c9a7ec41--
(2021-08-16T16:12:26.000000+09:00) First test
2022/1/19 13:06:41 - Find important mail!
1/22/1/19 13:06:41 - tweet_send
```

図 11 緊急通知に要する時間

起動で復元することができる。前の Slave は次の Slave の Master になることができ、Slave は他の Slave から接続や同期要求を受け取ることもできるので、Slave はチェーンの中で次の Slave の Master として機能する。次の Slave の Master になることで、Master への書き込み負荷を効果的に軽減することができる。途中で Slave のステアリングを変更した場合は、以前のデータをクリアして最新のものを再作成する。

4. システム評価

4.1 動作確認

動作確認を図 10 に示す。左側がユーザ情報入力システム（Web ページで A & K List データの入力、保存、読み出しなど）の動作確認を検証した。右側が IMAP エージェントで電子メールサーバからメールの情報読み出し、正常に重要メールを判別されたうえ、Find!!!!!! のフィードバックを表示した。

4.2 緊急通知に要する時間

緊急通知に要する時間を図 11 に示す。受信メールは以前のもの（届く時間 2021-08-16T16:12:26）だが、未読状態のため、本研究に影響がない。IMAP エージェント起動ですぐ（13:06:41）に A & K List データ読み出し、重要メールを検出され、同時にも Twitter にポストした。Twitter で確認できる時間は午後 1:06 であるが、秒数まではわからない。しかし重要メール判別システムにとって現段階では秒数がわからなくとも十分だと判断する。それぞれのステップをより明確にするため、表にまとめた。今研究は SMS、音声通話、LINE、Twitter の 4 つの方法について検討した。この中で、SMS と音声通話も Twilio の API を使用したが、Twilio は利用料がかかるため、今回は SMS と音声通話の動作確認は行わず、LINE と Twitter の動作確認のみを実施した。API の使い方は同じで、Twilio のサイトでも、API の使い方がわかりやすく紹介されているので、利用できると判断する。

```
*****Body is:
--_000_TY2PRO1MB4441B5FDC10B2E1CBE62520
Content-Type: text/plain, charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
This is the second test.
--_000_TY2PRO1MB4441B5FDC10B2E1CBE62520
Content-Type: text/html, charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
<html><head>
<meta http-equiv="3D"Content-Type" content="text/html"
ip>
```

IMAP エージェントプライマリ

IMAP エージェントセカンダリ

図 12 2つの IMAP エージェントの動作確認

4.3 フェイルセーフ機能の評価

IMAP エージェントプライマリは 1 秒ごとに確認メッセージを作成し、IMAP エージェントセカンダリは 1 秒ごとに確認メッセージの検出を行う。確認メッセージを検出することができないとき、IMAP エージェントプライマリがダウンした状態であると判別し、IMAP エージェントセカンダリは IMAP エージェントプライマリの機能をすべて継承する。IMAP エージェントプライマリと IMAP エージェントセカンダリが同時に起動中であることを図 12 に示す。

データベースのレプリケーションの評価を図 13 と図 14 に示す。

1 つ目は Master、2 つ目は Slave であり、2 台のサーバーで構成される環境を構築し、2 台は違う IP アドレスで実験され、データをレプリケートしていた。実際応用するときには、Master と Slave は異なる場所（停電などの災害を防ぐ）、異なるネットワーク環境（ネット障害を防ぐ）はさすが、制限があるため研究のテストや動作確認は異なるネットワーク環境だけで行った。これにより、システムやネットワークに障害が発生した場合の冗長性が確保され、性能面でも複数のサーバーに読み取り操作を分散させることができた。

5. おわりに

本研究では IMAP エージェントにより電子メールサー

