Vol.2015-DPS-162 No.30
Vol.2015-CSEC-68 No.30
2015/3/5

大規模災害を考慮したモバイルクラウド型災害情報共有システム

菊池瑶介[†1]　　柴田義孝[†2]

日本は地理的な条件により，自然災害が非常に発生しやすい環境に置かれている．2011 年 3 月 11 日に発生した東日本大震災では広域にわたる通信インフラの欠損，輻輳が発生した．このような事態から，障害や負荷変動を考慮したシステムインフラの重要性が明らかになった．本研究では，大規模災害を考慮したモバイルクラウド型災害情報共有システムを構築し，不安定な通信環境下においても止まることのないシステムインフラを実現することを目的とする．本システムでは，Web システムをクラウド基盤上で動作させ，サーバ群の障害や負荷変動にも柔軟に対処できるようにする．加えて，災害発生直後のような安定した通信が得られないような環境においても可能な限り迅速な情報共有を実現させるため，モバイルクラウドを導入し，被災地域の巡回による情報収集を可能にする．また，サーバ間の通信に DTN プロトコルを使用することにより，通信途絶時はデータを蓄積し，通信が回復したときにデータを送信するといった動作を実現させる．

Mobile Cloud Based Disaster Information System Considering Large-scale Disasters

Yosuke Kikuchi[†1]　　Yoshitaka Shibata[†2]

Japan is likely to occurring disaster from geographical conditions. For example, the Great East Japan Earthquake that occurred on March 11, 2011, communication infrastructure was damaged over a wide area. From such a situation, there is revealed importance of system infrastructure considering the failure or load change. In this study, we build a mobile cloud based disaster information system considering the large-scale disasters and we aim to infrastructure that does not stop under unstable communication environment. In this system, to operate Web system in the cloud infrastructure and to be able to deal flexibly to failures or load change of server group. In addition, we introduce mobile cloud to allow collecting information by the circulation of affected areas to achieve sharing information rapidly as much as possible in unstable communication environment such as after the disaster. Also, by using the DTN protocol to communicate between servers, the system achieves operation such as to store data when communication is interrupted, and to send stored data when communication is recovered.

1. はじめに

日本はその地理的条件から，災害が発生しやすい環境に置かれている．特に地震による被害は大きく，2011 年 3 月 11 日に発生した東日本大震災では甚大な被害を受け，広域にわたる通信インフラの欠損，輻輳が発生した． [9]近年ではこのような事態に備え，Web ベースの災害情報システムが提供され，災害発生後の情報収集，安否確認，意思決定・対応の迅速化に大きく貢献している．

現在はビジネスの効率化のため，クラウドサービスを導入しているユーザが増加している．クラウドサービスにより，ハードウェア・ソフトウェアの購入，導入・設定，メンテナンスといった一連の工程をデータセンタで一括管理することが可能である．また，ユーザ自身が物理的なサーバを持つ必要が無くなるため，システム運用にかかるコストの削減につながる．さらに，仮想化技術を利用することで，ユーザ自身によってプライベートクラウドを導入し，必要に応じて仮想サーバを構築することが可能となる．このように，クラウドサービスは Web システムを運用する上で多くのメリットが存在する．

一方，筆者らはこれまでに移動通信環境を考慮した分散型災害情報共有システム [1]の研究を行ってきた．このシステムはバックグラウンドでネットワークを監視し，データを送信可能な時に送信する機能を持つ．しかし，本システムはサーバに対する急激な負荷変動や障害を考慮していないといった問題がある．

災害時の情報システムの運用を想定した場合，外的要因（地震，津波など）によるサーバ，ネットワークの障害やサーバに対する負荷の集中といった点を考慮する必要がある．[10]

本研究では，大規模災害を考慮したモバイルクラウド型災害情報共有システムを構築し，不安定な通信環境下においても止まることのないシステムインフラの実現を目標とする．先行研究 [1][2]のシステムを動作させるためのクラウド基盤を構築し，サーバ群の障害や急激な負荷変動にも柔軟に対応できるようにする．また，複数の自治体に対して必要に応じてサーバ資源を動的に配分できるようにする．これにより，各自治体が持つリソースを最大限に活用できる．さらに，通信途絶時においても情報共有を実現させるため，DTN プロトコルを利用した通信を行う．加えて，本

†1 岩手県立大学大学院ソフトウェア情報学研究科
Graduate school of Software and Information Science, Iwate Prefectural University Graduate School
†2 岩手県立大学ソフトウェア情報学部
Faculty of Software and Information Science, Iwate Prefectural University

ⓒ 2015 Information Processing Society of Japan

システムではモバイルクラウドを導入し，移動型のクラウドによる被災地域の情報収集を行うことで，通信途絶地域における迅速な情報共有の実現を目標とする．

2. 先行研究と問題点

2.1 移動通信環境を考慮した分散型災害情報共有システム

これまでの先行研究のシステムでは，大きな遅延，頻繁なリンクの切断が発生するような環境においても災害情報システムの動作を実現している．バックグラウンドでネットワークを監視し，データを送信できるときに送信する機能を持つ．また，自治体毎に設置された災害情報サーバ間で円滑な情報共有が可能である．しかしながら，サーバに対する急激な負荷変動や障害を考慮していない，津波などの外的要因によって各自治体が持つサーバリソースが被害を受けた場合，それらのリソースを活用できないといった問題が挙げられる．

2.2 統一的な時系列表示を可能とする災害情報共有システム

電子地図上に被災地情報を重ねて表示することにより，視覚的に分かりやすく情報を確認することが可能である．また，登録する情報の種類によってアイコンや図形を使い分けることで，どのような被災地情報が登録されているか一目で判断することができる．さらに，シークバーを操作することにより，登録された被災地情報を時系列的に参照することが可能である．

3. システム概要

3.1 システム概要

本システムの概要を図 1 に示す．本システムは，主に Global Disaster Cloud，Temporary Server，Local Disaster Cloud によって構成されている．Temporary Server は各自治体があらかじめ持っているサーバと想定する．被災地域外では，システム管理者がクラウドサーバの管理を行う．また，自治体関係者，一般ユーザに対して災害情報閲覧に特化した機能が提供される．

被災地域内で安定したネットワーク環境を持つ地域では，主にモバイルクライアントから災害情報の登録を行う．不安定なネットワーク環境を持つ地域では，災害発生後に各自治体がそれぞれ Temporary Server を立て，地域ごとにローカルで情報の収集を行う．その後，Local Disaster Cloud を搭載した中継車が被災地域を巡回し，Temporary Server に蓄積された情報を収集し，Global Disaster Cloud に情報を届けることで通信途絶地域における情報収集を実現する．

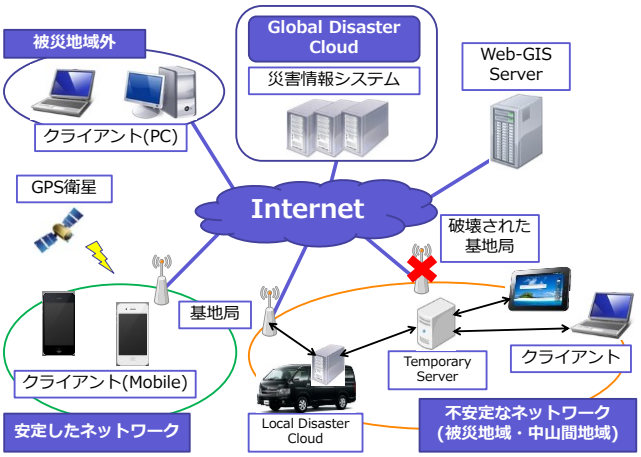

図 1. システム概要図

3.2 本研究におけるクラウド

- Global Disaster Cloud (GDC)

すべての地域の情報が集約されるクラウドを示す．各地域に設置された Temporary Server や Local Disaster Cloud に蓄積された情報を集約する．

- Temporary Server

各地域があらかじめ持っているサーバリソースを示す．通信が可能な場合，各地域の Temporary Server 間でサーバリソースの共有を行う．通信が不可能な場合においては，接続が回復するまでローカルでの災害情報の蓄積・共有を行う．このサーバは地震・津波などによる通信途絶，インフラの障害などの事態に対し，各地域が災害発生後に一時的に立てることを想定している．

- Local Disaster Cloud (LDC)

中継車に搭載することを想定した可搬型のサーバリソースを示す．各地域の Temporary Server に蓄積された情報を集約し，GDC に配信する．LDC の導入により，通信が途絶した状況においても GDC に可能な限り迅速に情報を配信させることで，情報伝達の遅れを最小限にすることが可能となる．

3.3 DTN (Delay Tolerant Networking)

DTN (Delay Tolerant Networking) [6][7]とは，通信が頻繁に切断したり，大きな伝送遅延が生じたりする劣悪な通信環境においても信頼性のあるデータ転送を実現する通信方式である．リアルタイム性を必要とする双方向通信には適さないが，低スループットかつ頻繁にリンクが切断するような不安定な通信環境においても通信が可能である．

本システムでは Temporary Server - GDC 間，LDC - GDC 間の通信に DTN プロトコルを用いる．これにより，Temporary Server および LDC は通信途絶時にはデータの蓄積を行い，通信が回復したときに蓄積したデータを GDC

ⓒ 2015 Information Processing Society of Japan

に配信することが可能となる．

3.4 システムの機能

● 仮想サーバ群のリソース制御機能

クラウド上で動作している仮想サーバの状態(CPU 使用率，RAM 使用率)を監視し，サーバリソースの自律的な制御を行う．具体的にはサーバ群への負荷分散，リソースの追加・削減といった操作を行う．

● サーバリソースの動的配分機能

各自治体の Temporary Server 間においてサーバリソースの共有を行い，リソースを動的に配分できるようにする．

● クラウド間におけるデータ同期機能

各自治体の Temporary Server や LDC によって収集された情報の同期を行い，収集された情報を共有できるようにする．

● DTN 通信機能

通信途絶時にはサーバ上にデータを蓄積し，通信回復時に蓄積されたデータをまとめて送信するといった動作を実現させる．

3.5 システムの動作(通常時)

通信が可能な場合におけるシステムの動作イメージを図 2 に示す．この場合，GDC から被災地情報の提供を行う．GDC にはすべてのサーバ群の情報が集約される．また，災害発生後のシステム障害やサーバに対する負荷変動に応じて，サーバリソースの動的な制御を行う．具体的には，リソースの追加・削除，負荷分散，ロードバランサのルール変更といった処理を行う．

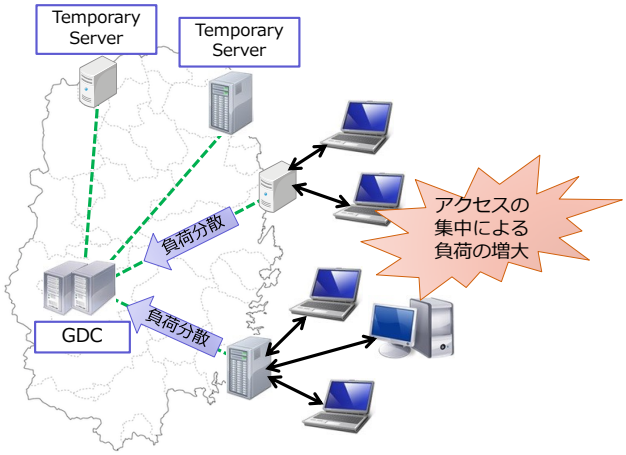

図 2. 通常時におけるシステムの動作

3.6 システムの動作(通信途絶時)

通信が不可能な場合におけるシステムの動作イメージを図 3 に示す．GDC にはすべての Temporary Server および LDC に蓄積された情報が集約される．LDC には 3G，WiMAX，LTE，IEEE802.11a/b/g/n/j などの無線通信装置が搭載されている．災害発生後に LDC を搭載した中継車が被災地域を移動し，各種クライアントから被災地情報の登録を行う．Temporary Server はインターネットに依存せずにローカルで被災地情報の収集を行う．その後，中継車が再度被災地域を巡回し，Temporary Server に蓄積された情報を収集していく．最後に，LDC に集約された情報を GDC へ送信することで GDC-Temporary Server 間，GDC-LDC 間での情報の同期を行う．ここでの通信には DTN プロトコルを用いることで，通信状態が劣悪な場合はデータを蓄積し，回復したときにデータを送信するといった動作を実現させる．

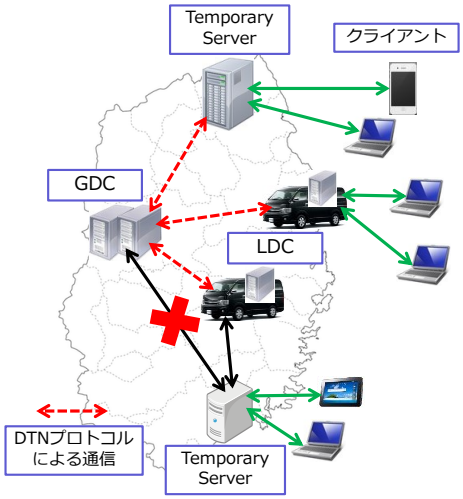

図 3. 通信途絶時におけるシステムの動作

4. システムアーキテクチャ

GDC，Temporary Server，LDC のアーキテクチャを図 4 に示す．それぞれのクラウドでは複数の VM 上で災害情報システムを動作させることによって災害情報の提供を行う．本システムは Monitoring モジュール，Resource Manager モジュール，VM Controller モジュール，DTN Transport モジュールによって構成される．

Monitoring は仮想マシンのリソース利用率(CPU, メモリ)を監視し，その結果を Resource Manager に送る．Resource Manager は Monitoring から送られてきた結果をもとに仮想マシン群のリソース制御を行い，VM Controller に仮想マシンの操作命令を送る．VM Controller は Resource Manager からの命令による仮想マシンの起動・停止・追加・削除といった操作を行う．DTN Transport では DTN プロトコルによるデータの送受信，蓄積などの管理を行う．

ⓒ 2015 Information Processing Society of Japan

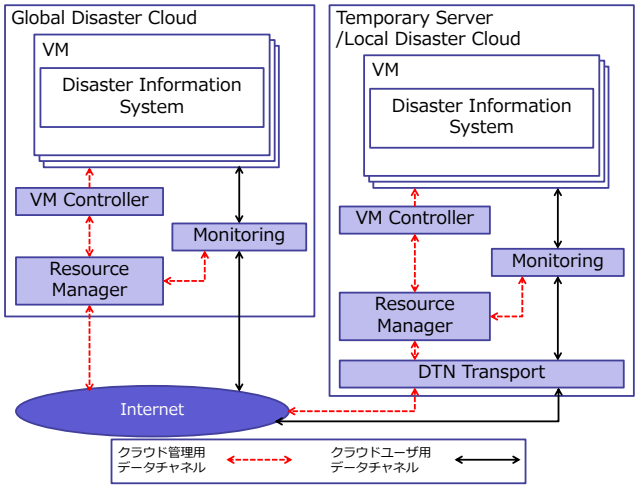

図 4. システムアーキテクチャ

5. サーバリソース制御アルゴリズム

5.1 サーバリソース増加処理

サーバリソース増加処理の流れを図 5 に示す．最初に Monitoring モジュールにより仮想マシン群のリソース利用率の監視を行う．次に，CPU およびメモリの使用率に応じたリソース量の調整を行う．CPU 使用率が X %以上の場合，仮想マシンに割り当てる CPU の周波数を U GHz 単位で増加させる．また，CPU 使用率が X %以下かつ RAM 使用率が Y %以上の場合，仮想マシンに割り当てるメモリの容量を V MB 単位で増加させる．これらを判断した後，リソース増加処理を実行する．最後に仮想マシンのスペックの変更に応じてロードバランサのルールを追加する． [3][4][5]

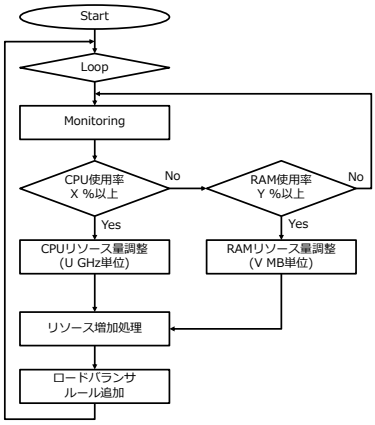

図 5. サーバリソース増加処理フロー

5.2 サーバリソース削減処理

サーバリソース削減処理の流れを図 6 に示す．増加処理と同様に Monitoring モジュールにより仮想マシン群のリソース利用率の監視を行い，CPU およびメモリの使用率に応じたリソース量の調整を行う．CPU 使用率と RAM 使用率がそれぞれ Z %以下，W %以下の場合，動作している仮想マシンを 1 台停止させる．これらを判断した後，リソース削減処理を実行する．最後に，停止した仮想マシンに対して割り当てられたロードバランサのルールを削除する．[3][4][5]

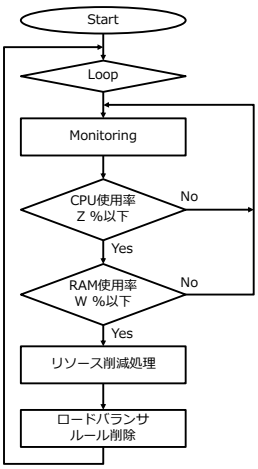

図 6. サーバリソース削減処理フロー

6. プロトタイプシステム

6.1 プロトタイプ環境

本システムのプロトタイプシステム構成を図 7 に示す．1 つのクラウドはクラウド全体の管理を行う Management Server とクラウドのリソースとなる Host で構成される．LDC は可搬性を考慮し，ラップトップ PC を使用している．GDC，Temporary Server，LDC のハードウェアスペックは表 1 に示す通りである．

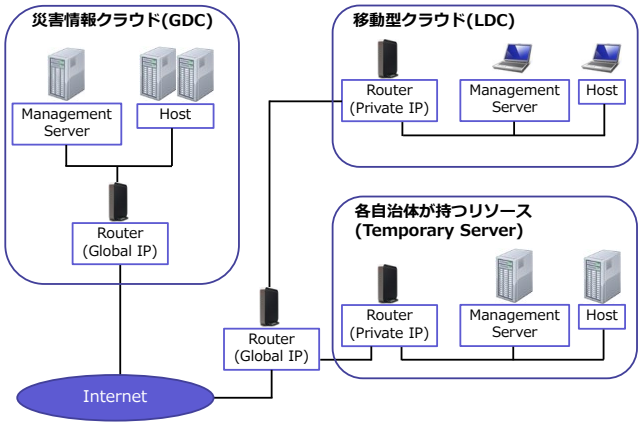

図 7. プロトタイプシステム構成

表 1. サーバのハードウェアスペック

	GDC, Temporary Server	LDC
CPU	Intel(R) Core(TM) i3-3240 Processor (3M Cache, 3.40 GHz)	Intel(R) Core(TM) i7-4712MQ Processor (6M Cache, 2.30 GHz)
RAM	4GB	16GB
HDD	500GB	1TB

6.2 LaDIPS (Large Disaster Information Portal Site)

被災地情報共有システムのインタフェースを図 8 に示す．電子地図上には複数のアイコンや図形が表示されており，

アイコンをクリックすることで画面右側に詳細情報が表示される．また，電子地図の上部に配置されているシークバーを操作することにより，被災地情報を時系列的に参照することが可能である．地図上にはさまざまな形式の情報を登録することが可能なため，多種多様な被災地情報を表現することができる．

本研究ではこのシステムをクラウド基盤上の仮想マシンで動作させ，サーバに対する負荷に応じて必要に応じてリソースの制御を行う．

図 8. 被災地情報共有システムのインタフェース

6.3 CloudStack

CloudStack [8]はオープンソースの IaaS クラウドソフトウェアの一つで，Amazon EC2 のような機能を備えたクラウド基盤をプライベートクラウド，またはパブリッククラウドとして構築することが可能である．

利用実績が多く，操作性に優れた GUI を備えているため，直近でのクラウドサービスの提供やプライベートクラウド環境の構築に適している．また，ロードバランサやファイアウォールなどの機能を標準搭載し，拡張性を考慮した内部アーキテクチャを持っているといった特徴がある．本研究では CloudStack を利用してクラウド環境を構築した．また，CloudStack API を使用し，仮想マシン群のリソース利用率に応じた動的リソース制御スクリプトの実装を行った．

7. システムの評価

7.1 システム評価項目

本システムの有用性を検証するため，以下の 3 点についてシステムの評価を行う．

- 不特定多数のユーザのアクセスにどの程度対処可能であるか
- 複数のクラウド間におけるリソース制御，負荷分散ができているか
- GDC，Temporary Server，LDC の間でデータの同期が実現できているか

7.2 予備実験

本プロトタイプシステム構成においてどの程度のアクセスに対処可能であるかを把握するため，予備実験として LDC の性能評価を実施した．本実験では被災地域の各自治体が VM を 1～5 台利用した場合を想定する．仮想マシンには LaDIPS がインストールされている．この仮想マシンに対して HTTP リクエストを生成し，そのときの平均応答時間，スループット(req/s)を計測した．実験結果はリクエスト間隔を 1 秒から 20 秒に変化させた場合における結果の平均値を示している．実験には JMeter [11]を使用した．実施条件および実験環境は以下の通りである．

- VM1 台のスペック
  CPU: 1GHz, RAM: 1GB, HDD: 10GB
- スレッド数(アクセス数)
  VM1 台: 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600
  VM2 台: 100, 150, 200, 250, 300, 350, 400, 450, 500
  VM3 台: 100, 150, 200, 250, 300, 350, 400
  VM4 台: 100, 150, 200, 250, 300, 350, 400
  VM5 台; 100, 150, 200, 250, 300
- スレッド生成時間
  10 秒
- リクエスト間隔
  1 秒，2 秒，5 秒，10 秒，20 秒

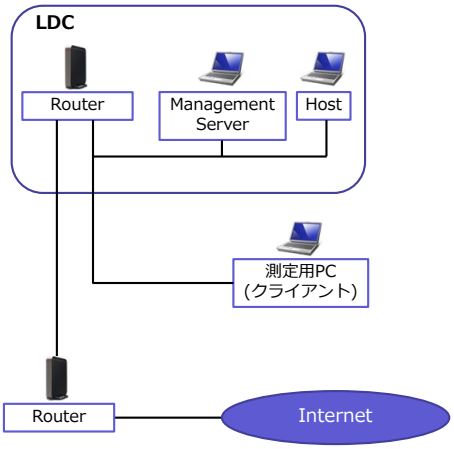

図 9. 実験環境

7.2.1 1 つの自治体が 1 台の VM を利用した場合

VM を 1 台動作させた場合における結果を図 10 に示す．応答時間は 500 スレッドまでは低い値を維持していたが，600 スレッドから急激に増加した．スループットは 550 スレッドまでは期待値通りであったが，600 スレッドのときに低下した．この結果から，VM を 1 台動作させている場合は約 55 リクエスト／秒までであれば概ね期待通りの性能を発揮することが分かる．

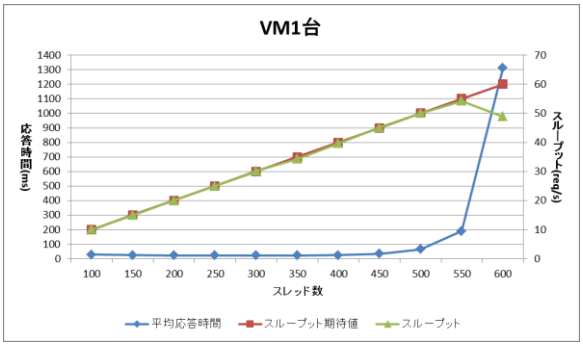

図 10. VM を 1 台動作させた場合における結果

7.2.2 2 つの自治体が 1 台ずつ VM を利用した場合

VM を 2 台動作させた場合における結果を図 11 に示す．応答時間は 450 スレッドまでは低い値を維持していたが，500 スレッドから急激に増加した．スループットは 450 スレッドまでは期待値通りであったが，500 スレッドのときに低下した．この結果から，VM を 2 台動作させている場合は約 45 リクエスト／秒までであれば概ね期待通りの性能を発揮することが分かる．

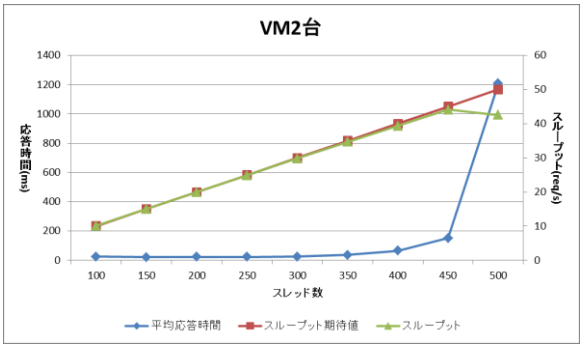

図 11. VM を 2 台動作させた場合における結果

7.2.3 3 つの自治体が 1 台ずつ VM を利用した場合

VM を 3 台動作させた場合における結果を図 12 に示す．応答時間は 250 スレッドまでは比較的速かったが，その後徐々に増加していった．スループットは 300 スレッドまでは期待値通りであったが，350 スレッドから低下していった．この結果から，VM を 3 台動作させている場合は約 30 リクエスト／秒までであれば概ね期待通りの性能を発揮することが分かる．

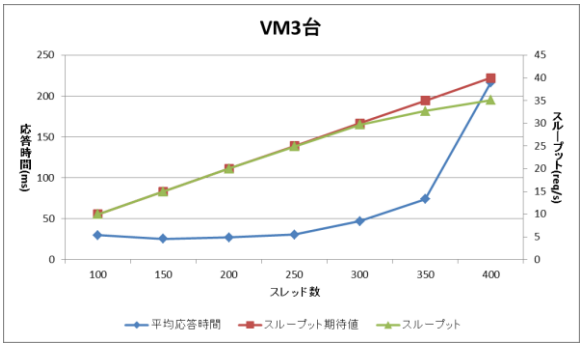

図 12. VM を 3 台動作させた場合における結果

7.2.4 4 つの自治体が 1 台ずつ VM を利用した場合

VM を 4 台動作させた場合における結果を図 13 に示す．応答時間は 250 スレッドまでは低い値を維持していたが，その後徐々に増加していった．スループットは 150 スレッドまでは期待値通りであったが，そこから 300 スレッドまで期待値より少し低下し，その後は徐々に低下していった．この結果から，VM を 4 台動作させている場合は約 15 リクエスト／秒までであれば概ね期待通りの性能を発揮することが分かる．

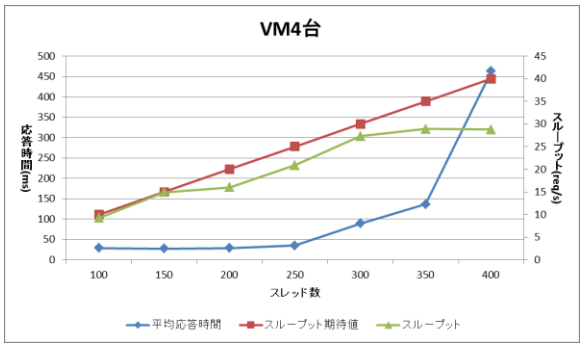

図 13. VM を 4 台動作させた場合における結果

7.2.5 5 つの自治体が 1 台ずつ VM を利用した場合

VM を 5 台動作させた場合における結果を図 14 に示す．応答時間はスレッド数の増加に応じて少しずつ増加していった．スループットは 150 スレッドまでは期待値通りであったが，その後はスレッド数の増加に応じて低下していった．この結果から，VM を 5 台動作させている場合は約 15 リクエスト／秒までであれば概ね期待通りの性能を発揮することが分かる．

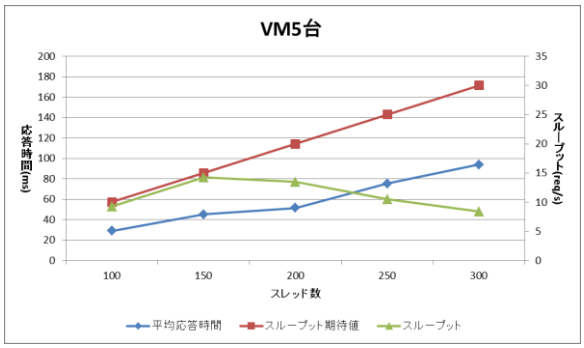

図 14. VM を 5 台動作させた場合における結果

7.2.6 実験結果に対する考察

本実験の結果から，VM の台数をある程度増加させて処理を行う方がより多くのリクエストを処理できることが証明された．実験結果から，VM が 1 台の場合は約 550 スレッド，2 台の場合は約 450 スレッド×2＝900 スレッド，3 台の場合は約 300 スレッド×3＝900 スレッド，4 台の場合は約 150 スレッド×4＝600 スレッド，5 台の場合は約 150 スレッド×5＝750 スレッドを期待通りの性能で処理可能であることが分かる．また，VM が 2 台，3 台の場合は 900

ⓒ 2015 Information Processing Society of Japan

スレッドなのに対し，4 台の場合は 600 スレッド，5 台の場合は 750 スレッドとなっていることから，VM を増やし過ぎると全体的な性能が低下することも証明された．

8. まとめ

本論文では，大規模災害を考慮したモバイルクラウド型災害情報共有システムを提案した．本システムにより，地震や津波などによるサーバ障害に対するリスクの分散，災害時の状況に応じた柔軟なサーバリソース制御が可能となる．また，モバイルクラウドの導入により，災害発生後の状況下においても被災地域の情報の迅速な収集を実現する．加えて，DTN プロトコルを用いた通信により，通信が途絶した場合はローカルでデータを蓄積し，通信回復時にデータを送るといった動作が可能となり，どのような通信状態においても被災地情報の共有を実現する．

今後はサーバリソース制御機能，クラウド間のデータ同期機能，DTN 通信機能の実装及びシステム全体を含めた性能評価実験を実施し，本システムの有用性の検証を行う．

参考文献

1) 佐々木 豊，柴田 義孝: 移動通信環境を考慮した分散型災害情報共有システムの構築，電子情報通信学会総合大会講演論文集 2011 年_基礎・境界，"S-65"-"S-66" (2011)
2) 佐々木 豊，柴田 義孝: 統一的な時系列表示を可能とする災害情報共有システムの構築，情報処理学会第 72 回全国大会，"3-427"-"3-428" (2010)
3) Trieu C. Chieu, Ajay Mohindra, Alexei A. Karve and Alla Segal, “Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment”, 2009 IEEE International Conference on e-Business Engineering (2009)
4) Gihun Jung, Kwang Mong Sim, “Agent-based Adaptive Resource Allocation on the Cloud Computing Environment”, 2011 International Conference on Parallel Processing Workshops (2011)
5) Zhang Zhang, Jizhong Han, Bo Li, Wei Zhou, Dan Meng, “Lynn: A Multi-Dimensional Dynamic Resource Management System for Distributed Applications in Clouds”, 2013 International Conference on Cloud and Service Computing (2013)
6) Delay Tolerant Networking Research Group, http://www.dtnrg.org/wiki/Home
7) Delay Tolerant Networking Research Group “DTN2 Documentation”, http://www.dtnrg.org/wiki/Dtn2Documentation
8) 日本 CloudStack ユーザ会, http://cloudstack.jp/
9) 総務省 ”東日本大震災における情報通信の状況”, http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h23/pdf/n0010000.pdf
10) 総務省 “通信における耐災害性の強化”, http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h24/html/nc134210.html
11) Apache Software Foundation “Apache JMeter”, http://jmeter.apache.org/

ⓒ 2015 Information Processing Society of Japan