

## RPC を用いた避難経路誘導システムの効率的な冗長化構成

多胡 樹<sup>†</sup> 滝本 宗宏<sup>††</sup> 神林 靖<sup>†††</sup>日本工業大学大学院電子情報メディア工学専攻<sup>†</sup> 東京理科大学理工学部情報科学科<sup>††</sup>  
日本工業大学先進工学部情報メディア工学科<sup>†††</sup>

## 1. はじめに

日本では、地震や津波、大雨による洪水、火山の噴火といった自然災害が頻発する。地震でいえば、2011年の東日本大震災(GEJE)、風水害でいえば、2019年10月に襲った台風19号が記憶に新しい。また、南海トラフ地震や首都直下型地震についても警告されている。

災害の発生時には、その状況によって適切な避難経路を選ばなければならない。例えば、家屋の倒壊や河川の氾濫によって、いつもは通れた道も突然通れなくなるかもしれない。そのような場合、われわれが開発している蟻コロニー最適化に基づく避難経路誘導システムを用いることによって、通れる道を探索することができる。さらに、ある地点の通行者数を基に、多くの人が通過した安全な道を優先的に案内することができる[1]。ここで、システムから提供されるデータは常に最新のものでなければならない。また、システムを動かしているマシンの耐故障性も考慮しなければならない。本研究では、移動エージェントを用いてマシンの故障に対処しながら、継続的に最新の情報を提供できるシステムを提案する。

## 2. 避難経路誘導システム

避難経路誘導システムについては、多くの研究が存在する。後藤らは、実効性のある避難経路誘導システムを構築するために、蟻コロニー最適化(ACO)アルゴリズムのフェロモン制御を避難経路に改良したものを提案した[2]。多賀らは、モバイルアドホックネットワークを介して、利用者が持つ端末同士で情報を共有するシステムを提案し、シミュレーションによってその有用性を検証した[3]。

本研究では、多賀らのシミュレーション結果に基づき、スマートフォンとサーバで構成されるサーバクライアントシステムを提案する。単純に、従来のサーバクライアントシステムと

して避難経路誘導システムを実現すると、サーバの故障によって、避難経路誘導システムとして機能しなくなるという単一障害点を生じる[1]。この単一障害点の解決に向けて、Zabbixを用いてシステムの構成を冗長化しようと試みた[4]。しかしながら、Zabbixがサーバの障害を検知するのに5分程度を要しており、避難経路誘導システムのような可用性を重視するシステムにおいて、Zabbixの利用は有用とはいえない。

## 3. システム概要

本研究では、先行研究で用いた Zabbix を、Python 3 による RPC で置き換えたシステム。RPCsMS(Remote Procedure Call's Monitoring System)を実現した。先行研究では、4台で基本最小構成単位としていた[4]のに対して、RPCsMSでは、基本最小構成単位として、サーバマシンと監視マシンの2台だけを使用する。監視を行うマシンとバックアップファイルを保存するマシンを、監視マシンとして集約させることによって、先行研究の半分の台数で実現している。

プログラムを実行中のサーバマシンで、不慮のトラブルが発生したと仮定する。RPCの接続が切れた時の例外処理を記述することによって、双方の接続が突然切れてもトランザクションを保持することができる。RPCの接続が切れた時に"connect out."と表示させるコードを図1に示す。

```
try:
    s = xmlrpc.server.
SimpleXMLRPCServer(("a.b.c.d", 8080))
    s.serve_forever()
except:
    print("connect out")
```

## 図1 トラブル発生を検出するためのコード例

この例外処理を有効に適用させることによって、マシンに不慮のトラブルが発生した時の対処法を用意することができる。例えば、Machine-Aをサーバマシン、Machine-Bを監視マシンとする。Machine-Aが故障したときの対処法は、監視をしていたMachine-Bがサーバマシンとして稼働することである。このとき、Machine-Bが故障したときの対処が必要になる。そこで、Machine-Bは、Machine-Aのデータを復元するとともに、次のサーバマシン候補でもあるMachine-Cに

## Providing Evacuation Support System Efficient Redundancy by Using Remote Procedure Calls

<sup>†</sup> Itsuki Tago, Department of Computer and Information Engineering, Nippon Institute of Technology

<sup>††</sup> Munehiro Takimoto, Department of Information Sciences, Tokyo University of Science

<sup>†††</sup> Yasushi Kambayashi, Department of Computer and Information Engineering, Nippon Institute of Technology

Machine-B の監視を依頼する．この繰り返しを行うことによって，RPCsMS の処理が継続するようにした．図 2 に概略を示す．

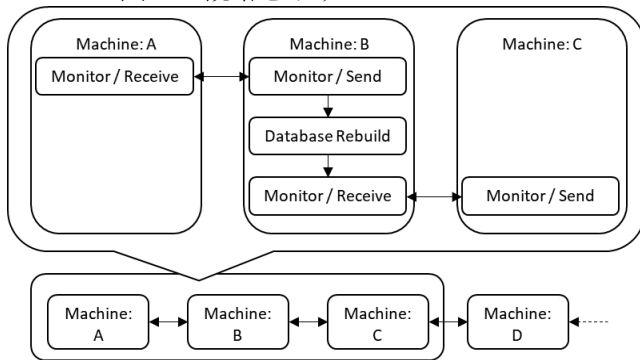


図 2 RPCsMS のシステム概要

#### 4. 検証実験

本研究の有用性を検証するために，2 台の Raspberry Pi 上に RPCsMS を実現した．2 台の Raspberry Pi をそれぞれ Machine-A, Machine-B とし，お互いに RPC を用いて繋がっている状態で，Machine-B から送られた Request を受け取った瞬間に Machine-A が故障することを想定した．このとき，使用したデータベースは，公開している避難所のデータや郵便番号のデータを含む 16.44MB のサイズであるとした．本実験では，1 日 100 回で 4 日間計 400 回の試行を行い，すべての回の復元時間を計測した．

400 回行ったうち，各復旧時間の頻度を図 3 に示す．図の柱状グラフが示すように，74.50 秒から 75.00 秒の頻度が最も高く 57 であり，平均値が 74.66 秒であった．これは，バックアップファイル 1MB 当たり 4.5 秒に相当する．Zabbix を用いた運用と比較して，2.6 倍の性能向上が認められた[4]．

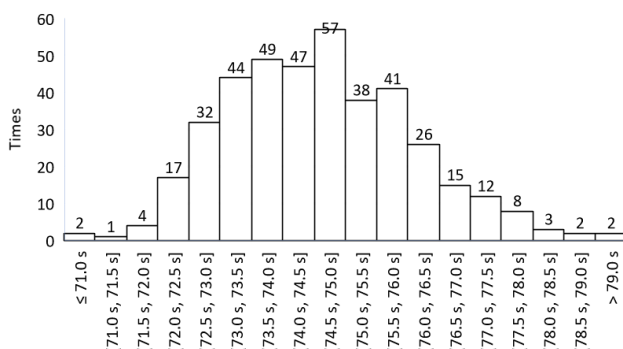


図 3 実験を 400 回行った結果

#### 5. 今後の展望

本論文では，Zabbix に変わる新たな監視システムを構築し，その有用性について検証を行った．新システムは，RPC を用いることによって，システムの異常を瞬時に発見できるようになり，当初想定していた目標時間より小さくなり，高い効果が得られた．また，システムを構成する

マシンを  $n+2(n>0)$  台にすることによって，数珠繋ぎの構成にしても問題なくバックアップが取れることを実験によって確認した．

本研究では，データベースを管理するマシンだけで復旧を確認したが，アプリケーションマシンとの連携がとれることを確かめる必要がある．さらに，今回構築したプログラムをもう一度見直すことによって，数ミリ秒ではあるが，バックアップにかかる実行時間を短縮できるのではないかと考えている．1 秒でも早くマシン移動が可能になることによって，被災者に対して常に最新のデータが提供できるようになる．

これ以外にも，避難所における避難人数と収容定員を把握し，それをリアルタイムに更新する必要がある．現在のシステムは収容定員 100 人を想定しているが，1000 人駆け込んでもおかしくない．最新の避難人数に更新できれば，定員に達しそうなときに，別の避難所へ誘導することができる．この更新のタイミングをどうすべきか，また避難所の場所の優先度はどのぐらいなのかを考慮するようしなければならない．

#### 謝辞

本研究は JSPS 科研費 JP17K01304, JP17K01342 の助成を受けたものです．

#### 参考文献

1. Kambayashi, Y., Konishi, K., Sato, R., Azechi, K., Takimoto., “A Prototype of Evacuation Support systems Based on the Ant Colony Optimization Algorithm”, Proceedings of Thirty-ninth International Conference on Information Systems Architecture and Technology, pp.324-333, (2018)
2. Goto, H., Ohta, A., Matsuzawa, T., Takimoto, M., Kambayashi, Y., Takeda, M., “A Guidance System for Wide-area Complex Disaster Evacuation Based on Ant Colony Optimization”, Proceedings of the Eighth International Conference on Agents and Artificial Intelligence, pp.262-268, (2016)
3. Taga, S., Matsuzawa, T., Takimoto., M., Kambayashi, Y., “Multi-Agent Approach for Evacuation Support System”, Proceedings of the Ninth International Conference on Agents and Artificial Intelligence, Vol.1, pp.269-274, (2017)
4. Tago, I., Suzuki, N., Matsuzawa, T., Takimoto, M., Kambayashi., Y., “A Proposal of Evacuation Support System with Redundancy Using the Mobile Agents”, Proceedings of the Thirteenth International KES Conference on Agents and Multi-Agent Systems: Applications and Technologies, pp.47-56, (2019)