M-006

障害発生時におけるサーバの自律的なサービス継続・復元手法の提案
A Proposal of Autonomous Service Continuation and Restoration Method of Server in Case of Failure

松舘 遼 † Ryo Matsudate　下村 祐人 † Yuto Shimomura　今井 信太郎 † Shintaro Imai　北形 元 ‡ Gen Kitagata

1 はじめに

現在，多くのサービスがネットワークを通じてユーザに提供されており，災害等でサーバやネットワークに障害が発生している場合においても，それらのサービスを継続して利用可能とすることは重要である．また，サーバやネットワーク管理者は，障害発生検知時に何らかの対応を行うことでサービスの継続を図るが，監視の負担が大きい，すぐに対応できる状況にない場合に対応が遅れてしまう等の問題がある．

以上の背景から本研究では，サーバに障害が発生した際の自律的なサービスの継続･復元手法の実現を目的とする．本稿では，手法の基礎的な仕組みを提案し，その効果を検証する．提案するシステムの機能には，サーバに障害が起こった際に他のサーバがサービスを代替するサービス継続機能と，最初にサービスを提供していたサーバが復旧してきた際にデータの更新を行いつつサービスを復元させるサービス復元機能の二つがある．本研究では，HTTP サービスと DNS サービスを対象とし，評価実験を実施する．

2 関連研究

2.1 ロードバランサ

HTTP サービスにおける障害対策方法として，サーバにかかる負荷を平等に振り分けるロードバランサの導入がある [1]．これを導入することで，障害が発生していないサーバのみにアクセスを振り分けることが可能となりサービスを継続することができる．しかし，高性能なハードウェアが必要になるため大きな導入コストがかかってしまう．

2.2 セカンダリ DNS

DNS サービスにおける障害対策方法として，セカンダリ DNS の設置がある [2]．こうすることで，プライマリ DNS サーバに障害が起こった際でもセカンダリ DNS が役割を担うことでサービスを継続することができる．しかし，この技術は DNS に特化した対策方法であり，様々なサービスには適用できない．

3 システム概要

本研究ではサービスを提供するサーバに障害が発生する場合を想定している．本研究におけるシステムは，正常時にサービスを提供するメインサーバ（以下「メイン」とする）1 台，サービスの代替処理やデータ更新処理などのシステム全体をコーディネートするコーディネートサーバ（以下「コーディネート」とする）1 台，メインのデータを定期的にバックアップしサービスを代替するバックアップサーバ（以下「バックアップ」とする）N 台 (N>0) で構成されている．前提として，コーディネートには障害が発生しないものとし，バックアップは 1 台以上が稼働しているものとする．また，各サーバ同士は ssh 接続のための鍵交換をしているものとする．

3.1 サービス継続機能

障害発生前に，コーディネートは定期的にメインに対して障害検知を行う．障害を検知した場合はブロードキャストで代替要求を送信する．代替要求を受け取ったバックアップは，代替可能であればコーディネートへ代替可能返答を送信する．コーディネートは，メッセージの応答が一番早かったバックアップにサービスの代替を指示する．指示を受けたバックアップは自身の IP アドレスをメインの IP アドレスに変える，サービスを起動するといった代替処理を行いサービスを引き継ぐ．図 1 にサービス継続機能のフローを示す．

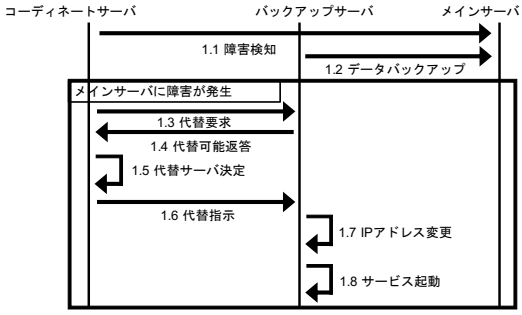


図 1: サービス継続機能のフロー

3.2 サービス復元機能

サービス代替後，コーディネートは定期的にメインに対して復旧検知を行う．復旧を検知した場合は，ブロードキャストで代替終了要求を送信する．代替終了要求を受け取った代替サーバは IP アドレスを本来の IP アドレスに戻す，サービスを停止するといった代替終了処理を実施する．その後，コーディネートは代替サーバのデータに基づきメインのデータを更新するためメインのサービスを停止させる．そして，代替サーバからデータをダウンロードし，そのデータをメインへ送信する．その後，メインのサービスを起動し，コーディネートはデータ更新完了確認をブロードキャストで送信する．図 2 にサービス復元機能のフローを示す．

† 岩手県立大学，Iwate Prefectural University
‡ 東北大学，Tohoku University

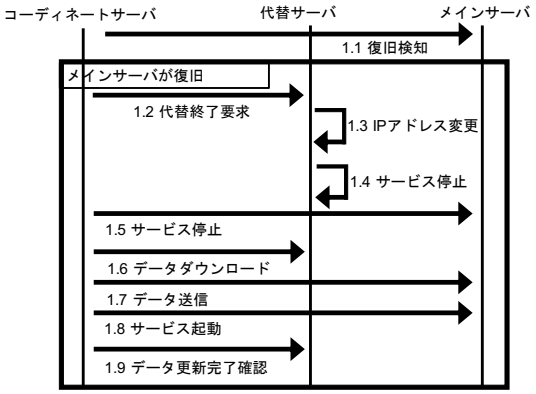
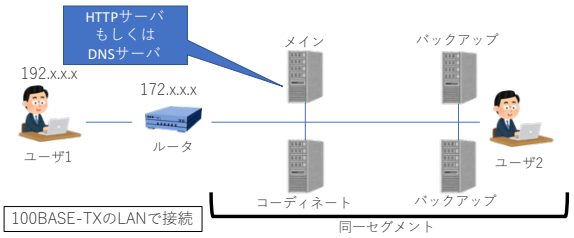


図 2: サービス復元機能のフロー

4 評価実験

4.1 実験環境

本研究における実験環境を図 3 に示す．本研究は，サーバに障害が発生するような状況におけるサービスの継続を想定している．そのため，メインとバックアップは本来分散配置されるべきであるが，今回はその基礎的な仕組みの検証のため，メインとバックアップは同一セグメント内に設置した．まず，HTTP サービスにおいてメインが提供する Web ページはテキストと画像 2 枚からなる合計約 2.3MB のページサイズとした．ユーザ 1 は wget コマンドを 5 秒ごとに実行することによりメインにアクセスする．続いて DNS サービスでは，ユーザ 2 が dig コマンドを 5 秒ごとに実行しドメインの取得可否を確認する．dig コマンドで取得したメッセージサイズは 115byte であった．またメインの LAN ケーブルの抜き差しによってメインにおける障害発生と復旧を再現した．



	OS	CPU	RAM	HTTPサービス	DNSサービス
1	Ubuntu	Intel core i5	4GB	メイン	コーディネート
2	Raspbian	Broadcom BCM2837	1GB	バックアップ	メイン
3	Raspbian	Broadcom BCM2837	1GB	バックアップ	バックアップ
4	Raspbian	Broadcom BCM2837	1GB	コーディネート	バックアップ
5	Mac OS	Intel Core i5	8GB	ユーザ1	-
6	Windows	Intel Core i7	16GB	-	ユーザ2

図 3: 実験環境

4.2 実験結果と評価

本実験では，障害検知と復旧検知の検知間隔を 1 秒から 10 秒に変えてユーザがデータを取得できるまでの時間を計測する．これを各検知秒数において 10 回試行しその平均時間で評価する．

図 4 と図 5 はそれぞれ HTTP サービスと DNS サービスにおける障害発生時の実験結果である．その結果，障害検知間隔が長いほどデータを取得できるまでの時間は長くなっているが，ユーザは継続してサービスを利用できることがわかった．

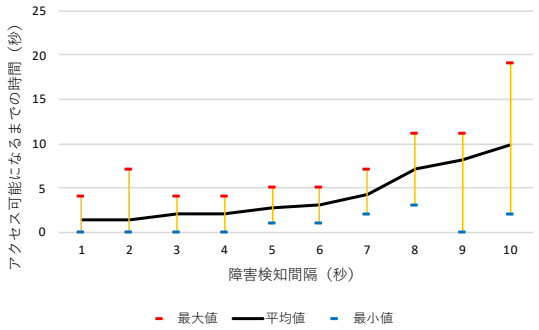


図 4: 障害検知間隔の変化によるデータ取得可能時間の推移 (HTTP サービス)

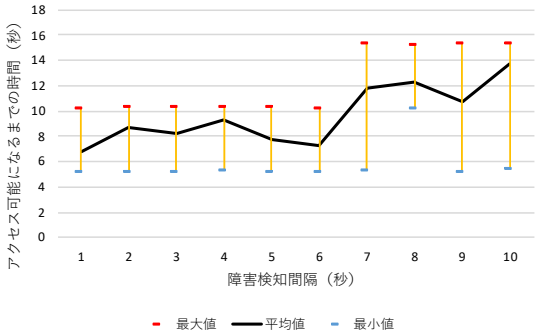


図 5: 障害検知間隔の変化によるデータ取得可能時間の推移 (DNS サービス)

次に復旧時においても同様に実験した．その結果，HTTP サービスにおいてユーザがデータを取得できるまでの時間は約 0.44 秒から約 1.25 秒，DNS サービスでは約 9.04 秒から約 29.73 秒であり，サービスによって多少の時間はかかるもののサービスの復元と継続が可能であることがわかった．

5 おわりに

本研究では，障害発生時におけるサーバの自律的なサービス継続・復元手法の実現を目的として，サービス継続機能とサービス復元機能を提案し，障害と復旧を再現し評価実験を実施した．その結果，障害発生時と復旧時それぞれにおいてサービスの継続と復元が可能であることがわかった．今後は，システムの改良を進めながら他のサービスにおいても実装を進めていきたい．

参考文献

[1] みやたひろし：サーバ負荷分散入門，pp. 1–15，ソフトバンククリエイティブ (2012).

[2] Albitz P., Liu C.（著），高田広章，小島育夫（監訳），小舘光正（訳）: DNS & BIND 第 3 版，オライリー・ジャパン，pp. 28–29（1999）.