2F-05

通信状態を用いた業務システム間の関連性分析方式の検討

今井 遼太郎† 白木 宏明† 大松 史生†
三菱電機株式会社 情報技術総合研究所†

1. はじめに

近年、企業の業務システムはより複雑に、より大規模になる傾向がある。業務システムに障害が発生した場合には、業務影響範囲を分析する必要があるが、現状では困難である[1]。業務影響範囲分析とは、ある業務システムを構成するサーバに障害が発生した場合に、他業務システムや他サーバへの影響を分析する技術である。

従来、業務影響範囲分析ではサーバ間の関連性を分析するために、構成管理データベースを構築する必要があった。構成管理データベースは、サーバからソフトウェアまたはハードウェアなどの様々な構成情報を収集し、包括的に管理することにより、構成情報間の関連性を理解することの支援を目的に導入される。しかし、構成管理データベースは、構成情報を包括的に管理するため、構築やメンテナンスにかかるコストが大きい。

そこで本稿では、サーバの通信状態を用いてサーバ間の関連性を分析することで、構成管理データベースを構築することなく、業務影響範囲分析を実現する手法を提案する。

2. 課題

一般に、システムで使用されるアプリケーションは複数存在し、アプリケーション毎に通信相手や通信内容、通信頻度は異なるため、サーバ間の関連性はアプリケーションに依存すると考えられる。つまり、あるシステムを対象とした業務影響範囲分析では、当該システムで使用されるアプリケーション毎にサーバ間の関連性を考慮する必要がある。したがって、アプリケーション単位での業務影響範囲分析が必要になると考えられる。

Study of the Relationship among Enterprise Information System using Communication Status
† Ryotaro Imai, Hiroaki Shiraki, Fumio Omatsu
Information Technology R&D Center, Mitsubishi Electric Corporation.

3. システム構成

図 1 に示すシステム構成において、あるシステムを構成する各分析対象サーバは通信状態を、業務影響範囲分析を実行する影響分析サーバは、プロセスとプロセスが属するアプリケーションの対応をまとめたプロセステーブルをそれぞれ保持している。

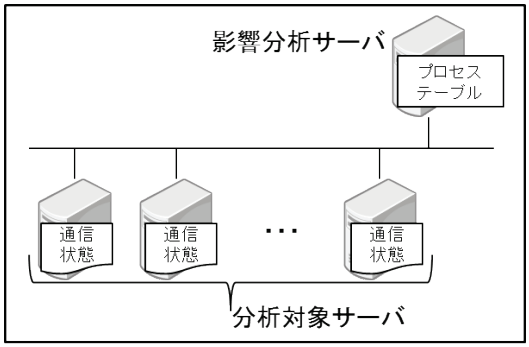

図 1：システム構成

本稿における通信状態とは、netstat[2]コマンドなどで取得したデータを示し、コネクションを構成単位とする。コネクションは、プロトコル名・送信元 IP アドレス・送信元ポート番号・宛先 IP アドレス・宛先ポート番号・接続状態・プロセス名を要素として持つ。ここで、接続状態とは、TCP コネクションの状態を意味する。

4. 提案手法

本手法には二つの目的がある。一つは、アプリケーション単位での業務影響範囲分析の実現である。もう一つは、サーバがアプリケーション中でどの程度中心的な役割であるかを相対的に判断する指標（以下、重要度）の計算である。サーバの重要度が大きいほど、サーバに障害が発生した場合の影響範囲が大きいことを意味する。

まず、アプリケーション単位での業務影響範囲分析手法について述べる。まず、各サーバから通信状態を取得する。取得した通信状態において、接続状態が ESTABLISHED であるコネクションのみを抽出して、各サーバからの通信状態を一つの通信状態に集約する。プロセステーブルにより、集約した通信状態をアプリケーショ

ン毎に分類する。アプリケーション毎の通信状態において、サーバを頂点、サーバ間のコネクションを辺、サーバ間のコネクション数を辺の重みとする隣接行列を計算する。このとき、隣接行列における要素が非 0 の場合、コネクションが存在するため、該当サーバ間には関連性が存在することを意味する。

次に、サーバの重要度の計算方法についてはPageRank[3]の考え方が適用できる。

① 隣接行列に対して、行と列を入れ替える転置隣接行列を計算する。
② 転置隣接行列に対して、各列の要素の総和で各列の要素を割る正規化転置隣接行列を計算する。
③ 正規化転置隣接行列の固有方程式を解き、絶対値が最大となる固有値を計算する。
④ 絶対値が最大となる固有値に対応する固有ベクトルに対して、要素の総和で要素を割る正規化固有ベクトルを計算する。

このとき、正規化固有ベクトルにおける各要素が対応する各サーバの重要度に相当する。

5. 処理事例

表 1 に示すプロセステーブルと図 2 に示す通信状態が与えられているとする。

表 1：プロセステーブル

プロセス名	アプリケーション名
foo.exe bar.exe	アプリケーション 1

```
 1 TCP AA.AA.AA.AA:a1  BB.BB.BB.BB:b1 ESTABLISHED foo.exe
 2 TCP BB.BB.BB.BB:b3 CC.CC.CC.CC:c2 ESTABLISHED bar.exe
 3 TCP AA.AA.AA.AA:a2  CC.CC.CC.CC:c1 ESTABLISHED bar.exe
 4 TCP BB.BB.BB.BB:b3 CC.CC.CC.CC:c2 ESTABLISHED bar.exe
 5 TCP DD.DD.DD.DD:d2  BB.BB.BB.BB:b4 ESTABLISHED foo.exe
 6 TCP AA.AA.AA.AA:a2  CC.CC.CC.CC:c1 ESTABLISHED bar.exe
 7 TCP BB.BB.BB.BB:b2 AA.AA.AA.AA:a3  ESTABLISHED bar.exe
 8 TCP CC.CC.CC.CC:c3 DD.DD.DD.DD:d1  ESTABLISHED foo.exe
 9 TCP BB.BB.BB.BB:b2 AA.AA.AA.AA:a3  ESTABLISHED bar.exe
10 TCP AA.AA.AA.AA:a2  CC.CC.CC.CC:c1 ESTABLISHED bar.exe
11 TCP BB.BB.BB.BB:b2 AA.AA.AA.AA:a3  ESTABLISHED bar.exe
12 TCP DD.DD.DD.DD:d2  BB.BB.BB.BB:b4 ESTABLISHED foo.exe
```

図 2：通信状態

通信状態は 12 コネクションから構成される。例えば、最初のコネクションは、TCP プロトコル、送信元 IP アドレスが AA.AA.AA.AA、送信元ポート番号が a1、宛先 IP アドレスが BB.BB.BB.BB、宛先ポート番号が b1、接続状態が ESTABLISHED、プロセス名が foo.exe であり、アプリケーション 1 における通信であることを意味する。送信元 IP アドレス、宛先 IP アドレスが 4 種類登場しているため、対応するサーバを A、B、C、D とするとき、隣接行列$\boldsymbol{A}$は式(1)で与えられる。

$$\boldsymbol{A}=\begin{pmatrix}0&1&3&0\\3&0&2&0\\0&0&0&1\\0&1&0&0\end{pmatrix} \quad (1)$$

式(1)から、サーバ A に障害が発生した場合、サーバ B やサーバ C が影響を受け、サーバ B に障害が発生した場合、サーバ A やサーバ C が影響を受けることを意味する。

次にサーバの重要度を求める。$\boldsymbol{A}$に対して、正規化転置隣接行列$\overline{\boldsymbol{A^T}}$は式(2)で与えられる。

$$\overline{\boldsymbol{A^T}}=\begin{pmatrix}0&3/5&0&0\\1/4&0&0&1\\3/4&2/5&0&0\\0&0&1&0\end{pmatrix} \quad (2)$$

$\overline{\boldsymbol{A^T}}$の固有方程式の解で、絶対値が最大となる固有値は$\lambda=1$であり、$\lambda=1$に対応する正規化固有ベクトル$\overline{\boldsymbol{P}}$は式(3)で与えられる。

$$\overline{\boldsymbol{P}}=\begin{pmatrix}12/61\\15/61\\17/61\\17/61\end{pmatrix} \quad (3)$$

式(3)から、アプリケーション 1 におけるサーバ A、サーバ B、サーバ C、サーバ D の重要度は 0.197、0.246、0.279、0.279 と計算される。

6. おわりに

本稿では、サーバから取得した通信状態を用いてアプリケーション単位での業務影響範囲分析手法とサーバの重要度の計算手法を提案した。本手法では、プロセステーブルを用いて通信状態を分類したが、サーバとサーバが属するシステムの対応をまとめたシステムテーブルを影響分析サーバに与えることにより、システム単位での業務影響範囲分析も実現できる。つまり、隣接行列に対して、何を頂点とするかで様々な単位での業務影響範囲分析が実現できると考えられる。これは、重要度についても同様である。

今後は、実データを用いて検証・評価を行い、本手法の有用性を示す。

参考文献

[1] 独立行政法人情報処理推進機構 “情報システム障害の再発防止のための組織的マネジメントの調査 WG 報告書”(2012)
[2] Linux Foundation「net-tools」<http://www.linuxfoundation.org/collaborate/workgroups/networking/net-tools>(2015 年 12 月 21 日アクセス)
[3] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web(1998)