

提案手法：サービス機能の実行及び配置

- サーバを一つのタブ空間としてモデル化
- サーバでのサービス機能の実行を化学反応式で記述
 - 反応の速度は反応物の濃度に応じて決定
 - サーバ資源の制約は酵素触媒反応のモデルを用いて記述

使用しているサーバ資源を表す物質

$$SERV + REQ + CATAL \xrightarrow{k_{12}} MEDATE$$

利用可能なサーバ資源を表す物質

$$SERV + REQ + CATAL \xrightarrow{k_{11}} MEDATE$$

$MEDATE \xrightarrow{k_{21}} SERV + REQ + CATAL$

$MEDATE \xrightarrow{k_{22}} SERV + REQ + CATAL$

$SERV \xrightarrow{k_{31}} 0$

$SERV \xrightarrow{k_{32}} SERV$

リクエスト量に応じてサーバ資源を使用してリクエストを処理

サービス機能が多く行われるとそのサービス機能の需要が多いと判断

サービス機能が行われないと、そのサービス機能の需要が少ないと判断

リクエストを持つサーバにサービス機能を移動

需要が多いサーバにサービス機能を配置

7 タブル空間(サーバ) サービス機能の実行結果を表す物質

提案手法：勾配場によるリクエストの移動

- リクエストが、より適切なサーバへ移動する挙動を化学反応式として記述
 - サーバの利用可能な資源とサービス機能に対する需要に応じて、各サービス機能の勾配場を形成

$SERV + CATAL \xrightarrow{k_{12}} SERV + CATAL + GRAD$

$GRAD \xrightarrow{k_{22}} GRAD + (GRAD)$

$GRAD \xrightarrow{k_{21}} 0$ $REQ \xrightarrow{k_{31}} REQ + (GRAD)$

SERVとCATALの濃度の積に応じてGRADを生成、GRADの濃度勾配場を構築

リクエストはGRADの濃度が高いサーバへ移動

タブ空間(サーバ)

濃度勾配

2015/2/17

8

性能評価：評価環境

- SINETのバックボーンネットワークポロジを用いた評価
 - 各ノードにサービス機能を提供するためのサーバが一台存在すると仮定し、提案手法を適用
 - 各サーバは単位時間あたり50のリクエストを処理できる資源を持つと仮定
 - CATAL(触媒)の初期濃度1,000に相当
- 確認する動作
 - サービス機能間の自動的なサーバ資源割り当て
 - 需要が多いサーバへのサービス機能の自律的な再配置

9 2015/2/17

シナリオ1：サーバ資源の割り当て

- 提供するサービス機能: 2種類
- ノード1の各SERVの初期濃度: 2,000
- リクエスト投入速度
 - リクエスト1: 単位時間当たり10
 - リクエスト2: 単位時間当たり30
- シミュレーション結果
 - リクエスト投入速度に応じて各サービス機能のMEDIATEの濃度が一定値に収束
 - CATALの濃度がMEDIATEの濃度の増加に応じて減少

サーバ資源をリクエスト量に応じて各サービス機能に割り当て、サービス機能を実行している

10

シナリオ2：サービス機能の再配置

- ノード8のSERVの初期濃度: 3,000
- ノード1へのリクエスト投入速度
- リクエスト: 単位時間当たり40
- シミュレーション結果
 - SERVがノード8からリクエストが投入されているノード1に移動している

サービス機能を実行するに適したノードにそのサービス機能が移動

サービス需要の時間による変化

11

シナリオ3：負荷分散

- ノード6のSERVの初期濃度: 1,000
- ノード3へのリクエスト投入速度
- リクエスト: 単位時間当たり80
- SERVの拡散範囲をノード4、5、6に制限
- ノード3ではサービスを実行できない
- シミュレーション結果
 - サービス機能の実行が三つのノードで分散的に行われている

負荷分散の実現

十分に時間が経過した際のサービス機能実行結果の生成速度

12 2015/2/17

提案手法の拡張：NFVに対する適用

- ▶ Network Function Virtualization (NFV)
 - ▶ ネットワーク機能が仮想化され、サーバに配置
 - ▶ フローは決められた順序で各機能を適用
- ▶ ネットワーク機能の配置場所やフローがどのサーバで機能を適用するかを決定するために提案手法を拡張

フローを適用する機能の勾配場に応じて移動

機能を適用後、フローは次に適用する機能のリクエストを持つフローに変化

フローの決定と機能の配置を自律的に行う

```

SERV1 | FLOW1 → SERV2 | SERV2 | FLOW2 → SERV3 | SERV3 | FLOW3
SERV1 | FLOW1 → 0 | SERV2 | FLOW2 → SERV3 | FLOW3 → FLOW3 (GRAD3)
SERV1 | FLOW1 → SERV1 | GRAD1 | GRAD1 → 0 | GRAD2 | GRAD2 → GRAD2 (GRAD2)

```

▶ 13

2015/2/17

まとめと今後の課題

- ▶ 仮想化ネットワークシステムにおいて、各サーバの自律分散的な動作によってサービス空間を構築する手法を提案
- ▶ シミュレーション評価により、リクエスト量に応じたサーバ資源割り当て、サービス機能の分散実行、適切なサーバへのサービス機能の再配置が実現できることを示した
- ▶ 提案手法のNFVに適用する拡張モデルを説明し、提案手法が現実的なアプリケーションに対応できることを示した
- ▶ 今後の課題
 - ▶ NFVに適用した拡張方式の性能評価
 - ▶ システム障害や環境変動に対する性能評価

▶ 14

2015/2/17