

# 小口研究室 研究紹介 (2016年度)

## (お茶の水女子大学理学部情報科学科)

### OpenStack環境下におけるデータベースの管理システムの検討 (研究担当:西出 彩花)

#### 研究背景

クラウドシステムの普及に伴うデータ転送量の増大

- DCでの事故等によるデータ損失の恐怖
- ハイブリッドクラウド上でのデータ転送時の精度の悪化
- セキュリティ面の不安
  - システムの切り替え(緊急時用、通常時用など)
  - 重要なデータを失わないための遠隔バックアップ
- 手動での制御の限界
  - 自動化アルゴリズムの導入

#### 研究概要

クラウド環境における災害時のデータ保護/管理手法を検討

- 外部から取り込んだ情報をもとに制御
  - DBアクセス処理を実行するアプリとDBそのものを別々に扱う
  - DBアクセスを行うサーバを遠方にマイグレートし、DBアクセス処理を再開して実行
  - DB: 冗長的に遠隔地にバックアップ

#### 基本性能検証

構築したクラウド環境と物理サーバ上で、TPC-Wベンチマークを用いて実験を行う

Pangeaを介してクエリを送信し、レスポンス時間とスループットを計測

**Pangeaの性能検証**

- スループット値
  - クラウド上: 78.24
  - DBを分ける: 122.1
  - 約36%の向上
- レスポンス時間
  - 確認できた事
    - DBを物理サーバに置く事でクラウド上でもPangeaの利用が有効に

#### 提案手法

クラウド上で通常通りにマイグレーションを行った場合と、提案手法を用いてマイグレーションを行った場合の所要計測時間の比較

→ 提案手法導入によるオーバーヘッドはほとんどない

#### 実験方針

- ハイブリッドクラウドの仮想的なシステムを手元で構築
- OpenStack (Icehouse): クラウド構築ソフトウェア
- Pangeaを用いて仮想マシン内のDBを冗長的に同期
- OpenFlowを用いて仮想マシンマイグレート時のトラフィックを制御
- クラウド上で動的制御

#### 提案手法

① 震源地  
② マグニチュードから場合分け

例) A. 震源地 岐阜県飛騨地方 & M7  
B. 震源地 茨城県西部 & M7  
C. その他の地域

災害情報検出 → Twitter Stream Reader → 同クラウド内別ノードに転送 → 別のクラウドに転送 → 自クラウド内で再バックアップ

#### 実験

提案システム上で、様々なマシンをクライアントとし、Pangeaを介してクエリを送信したままマイグレーションを行った場合の所要時間を計測する

クエリを投げながら転送を行うと、転送に時間がかかる。これは仮想マシンの大きさに比例する。

ある一定のクエリの量を超えると、それ以上転送速度の低下は起こらない

#### まとめ

緊急災害時のデータ損失対策

- 災害時のVMの自動転送
  - 自動的に対応を振り分ける
- 通常時の遠隔地でのDB同期
  - OpenStack上で検証
  - 提案システム実装によるオーバーヘッドは少ない
  - クエリを投げながら行うと転送性能が低下
  - ある一定量のクエリを送ると性能低下しなくなる

#### 今後の課題

- 転送を行うサーバ間の距離について
  - Pangea\*\*の利用
- セキュリティについて
  - Pangeaを改良して、バケットに危機管理レベルを追加することで管理レベルによって扱い方を変える

### 大規模災害時におけるSNSによる集合知に基づいたネットワークQoS制御のための情報検索 (研究担当:丸 千尋)

#### 研究背景

- 東日本大震災時には、ネットワーク機器からの情報だけで、ネットワーク全体の状況を迅速に把握することが困難
- 緊急時にはユーザがネットワークの状態に強い関心を寄せ、その情報をSNSを通して積極的に発信

#### 提案手法

従来の手法 - ネットワーク機器内で監視

提案手法 - ネットワーク機器外部の情報を利用

Twitterによる集合知に基づいたネットワーク制御を自動的/自律的に行うネットワーク障害検知システムの開発

#### 市区町村ごとのユーザが通信障害と思う程度の決定

平常時のツイートの頻度の分布をもとに緊急時の障害の程度を決定

平常時(何も障害が発生しなかった日)の30日分のツイートを1時間ごとに分割それぞれのファイルについて、市区町村ごとに

$$R = \frac{\text{ある市区町(ここでは仙台)を含むツイートの総数}}{\text{地名情報を含むツイートの総数}} \times 100$$

左図の縦軸は、全ファイルにおけるRの出現頻度

市区町村ごとに確率分布を正規分布にフィッティングし、 $\mu + 2\sigma$ と $\mu + 3\sigma$  ( $\mu$ :平均、 $\sigma$ :標準偏差)を求める

地震発生後60分間の通信障害に関するツイートに対して $R_{Emergency}$ を求める

$R_{Emergency}$ と $\mu + 2\sigma$ 、 $\mu + 3\sigma$ を比較し、ユーザが通信障害と思う程度を決定

#### 提案手法

市区町村単位で通信障害に関する情報を検出

SNSの集合知から検出した細粒度の地理的な障害情報をもとに、ネットワーク制御を実施

ツイートの収集 - ブートストラップ法を用いて通信障害に関するキーワードを決定し、そのキーワードを含むツイートを取得

候補データ抽出処理 - キーワード検索で取得したツイートを市区町村単位でまとめる

地名分類処理 - 地名を「その地名で通信障害が発生しているのか」「その他の場合であるのか」に分類

ネットワーク制御 - ネットワーク障害検知システムからの出力結果をもとに、ネットワークの経路制御を実施

SNSの集合知に基づいたネットワーク制御を実現するために、自動化システムを構築

#### 評価実験

東日本大震災、2016年に発生した熊本地震のツイートコーパスを用いて、全ての市区町村を評価

- 東日本大震災時の岩手県と宮城県の結果について、市区町村単位で評価
- 提案手法が検出した11市区町村のうち8市区町村を正確に検出
- 検出できなかった市区町村については、そもそも元のツイートにその市区町村名が出現していなかったことが原因
- 他の情報源を組み合わせて利用することで解決できる

#### 今後の課題

電話網以外のネットワーク(メール等)の障害検知

Twitter以外の情報源を組み合わせて利用

### 災害時におけるSNS情報を活用したプログラマブルなQoS制御システムの実装と評価 (研究担当:柳田 晴香)

#### 研究背景

東日本大震災時NWが断絶し、通信出来ない事象が発生

<原因>

- 人の手による制御の限界
- ネットワーク全体の状態を迅速に把握することが困難

SDN (Software Defined Network)による自動集中制御

- SNS解析による迅速な状態把握
- 災害時に最適な帯域制御

上記により、災害時にもユーザが安心して連絡を取り合えるNWの構築を目指す

#### 提案システム

- アプリケーションの識別
- SNS解析による障害情報の受理
  - リアルタイムにツイートを解析し、通信障害に関係のあるツイートを同研究室のシステムより取得
- スイッチ間のリンクのコスト値更新
  - 初期値を1とし、障害ツイートの中にスイッチと対応させた地名を含むツイートが20件以上あったら、60秒毎に+1
- ダイクストラ法によるコスト値最小の経路探索
- REST-APIでスイッチに経路設定、帯域設定
- アプリケーション毎のQoS制御

#### FLARE実験実装

FLARE switch1	CPU	Core i7-3612QE Mobile
FLARE switch4	Memory	8GB
	OS	CentOS 6.4
h1 ~ h4	CPU	Core i5-4210 M 2.6GHz
	Memory	8GB
	HDD	SATA 500GB 5400RPM
	OS	Ubuntu14.04
h5, h6	CPU	Xeon E3-1241 v3 3.5GHz
	Memory	8GB
	HDD	SATA 1TB 7200RPM
	OS	Ubuntu14.04

h1 ~ h4: 1Gbps, h5 ~ h6: 10Gbps

h4はProxyサーバ、h6はWebサーバ

【前提】

- 災害時に、Skypeなどの連絡用アプリを優先
- YouTubeなど帯域を食うアプリを制限 (YouTubeはファイルダウンロードで代用)
- s1~s4それぞれ、岩手、東京、京都、福岡に対応
- h1がスタートでh4がゴール

【シナリオ】

- 提案システムなし
- 提案システムあり (経路制御)
- 提案システムあり (帯域制御)

#### ①提案システムなし

- FLARE switch1~FLARE switch4をそれぞれ、岩手、東京、京都、福岡に対応
- 岩手付近のh1がスタート
- 東京付近のh4がゴール

2011年3月11日14時から15時の実際のツイート

- ツイートより、岩手・東京間で障害検知
- 障害前後でも、SkypeとYouTubeを経路1に

スループット測定

#### ②提案システムあり (経路制御)

2011年3月11日14時から15時の実際のツイート

- ツイートより、岩手・東京間で障害検知
- YouTubeのみ、京都を通る迂回経路への自動経路切替に成功

スループット測定

#### ③提案システムあり (帯域制御)

障害後は、帯域制御でYouTubeに制限かける

スループット測定

まとめ

- 提案手法を用いることで、RTTを20-30ms程度に抑えつつSkype (情報伝達のためのアプリケーション)のスループット値を約800Mbps上げることに成功
- YouTube に関してスループット値の向上が見られることから、本提案システムによりネットワーク全体のスループット値の向上にも成功