小口研究室　研究紹介　(2013年度)
(お茶の水女子大学理学部情報科学科)

外部情報の取得に基づいたネットワークトラフィックの最適化
（研究担当：原瑠理子）

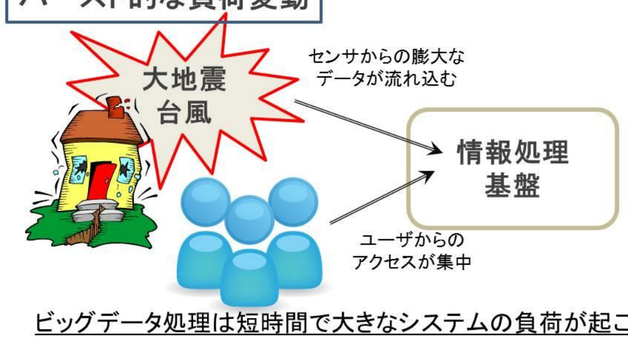
研究背景

- センサ技術の普及
- 携帯型デバイスの発展
- クラウドやDCの増加

⇩

ビックデータへの対応が重要となってきた

バースト的な負荷変動



ビッグデータ処理は短時間で大きなシステムの負荷が起こる

研究目的

従来のネットワーク制御

- 一般的に緩やかな負荷変動を念頭に検討されている
- 短時間で激しく変化するバースト的な負荷変動に追従することは難しい

OpenFlowを用いた投機的なネットワーク制御を提案

負荷変動を外部情報から予測

⇩

OpenFlowによるネットワーク制御

⇩

システムの再構成

自然災害など不測の事態への対応

実験概要

外部情報の取得

気象庁がTwitter上で発信している緊急地震速報をモニタリングし、地震情報を取得

本研究では，疑似的に緊急地震速報と同じ内容のツイートを流して実験を行う

まとめと今後の課題

負荷変動を外部情報から予測して、問題解決のための制御モデルを提案し、その制御モデルの動作を確認した

今後の課題

- 制御した後の復帰のタイミングの検討
- 計算機資源の最適化の検討
- アプリケーションによる制御
  → アプリケーションごとに優先順位を付ける

OpenFlowによる制御モデル

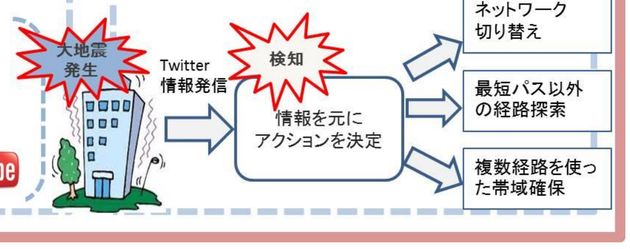
OpenFlowコントローラフレームワークの1つであるTremaを使って制御を行う

3つのシナリオを提案

(1)マグニチュード6.0以上の地震を観測
⇒ **ネットワークの切り替え**

(2)ある特定の地域でマグニチュード6.0以上の地震を観測
⇒ **最短パス以外の経路探索**

(3)マグニチュード7.0以上の地震を観測
⇒ **マルチパスによる帯域確保**



ハイブリッドクラウドにおけるデータベース同期手法
(研究担当：細谷　柚子)

研究背景

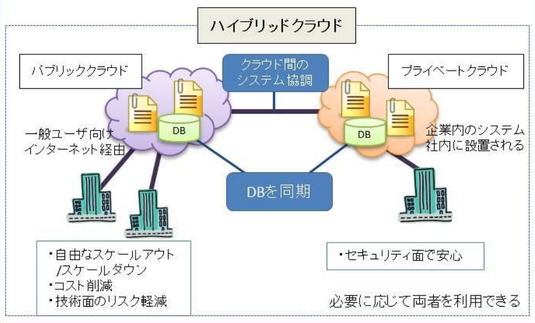
クラウドコンピューティングモデルの出現
⇒パブリッククラウド、プライベートクラウドが普及
2つのクラウドを併用する：ハイブリッドクラウド

⇩

ハイブリッドクラウド
⇒必要に応じた両者の使い分け
⇒両者のシステム協調が必要

⇩

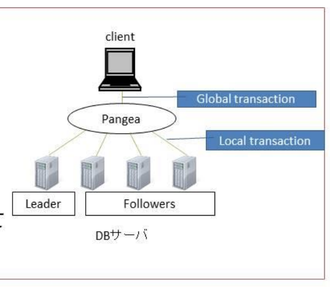
システム協調
⇒重要度の高いデータベースサーバを協調させるべき



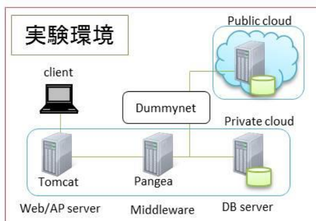
Pangea　(NTT三島)

LAN環境を前提としたデータベース同期ミドルウェア
・スナップショット分離性
・同期複製
・DBサーバの修正不要

クライアントからPangeaを介してDBサーバにアクセス、同期



実験環境



Dummynet:
16 ms (東京－大阪間)
256 ms (日本－日米/日欧間)

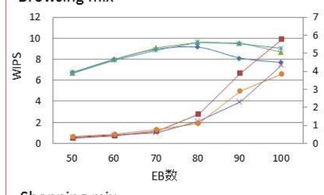
ベンチマークTPC-W
・Webサーバでの DB処理を評価
・仮想的なブラウザ(EB)が トランザクション処理を要求
3種のワークロード

ワークロード	Read-only	Update
Browsing mix	95%	5%
Shopping mix	80%	20%
Ordering mix	50%	50%

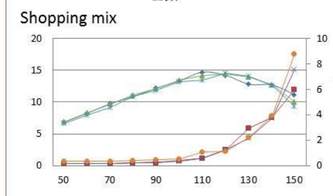
スループットWIPS
(Web interactions per second)
1秒当たりのトランザクション数
レスポンス時間 WIRT
(Web interaction response time)

性能評価

Browsing mix

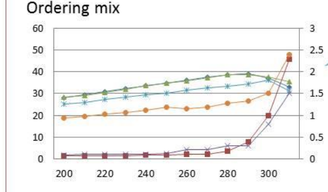


Shopping mix



遅延による大きな性能低下はみられない

Ordering mix



高遅延時に性能低下がみられる

Ordering mixの高遅延の場合に性能低下
ースループットの悪化
ーレスポンス時間の増加

⇓

Pangeaを修正し
レスポンス時間を短くする

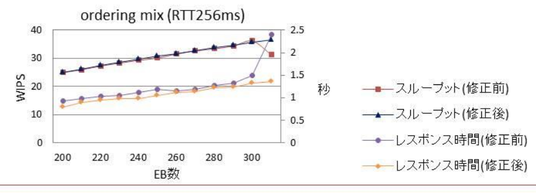
提案手法

重い書き込みを伴うCOMMIT処理を修正
クライアントに応答を早く返す
⇒レスポンス時間が短くなると予想

修正前プロトコル	
1	クライアントからPangeaにリクエスト送信
2	PangeaからLeaderにリクエスト送信
3	Leaderから応答が返る
4	全Followerにリクエスト送信
5	全Followerから応答が返る
6	Pangeaからクライアントに応答が返る

修正後プロトコル	
1	クライアントからPangeaにリクエスト送信
2	PangeaからLeaderにリクエスト送信
3	Leaderから応答が返る
4	Pangeaからクライアントに応答が返る
5	全Followerにリクエスト送信
6	全Followerから応答が返る

提案手法の性能評価



まとめと今後の課題

- Pangeaの基本性能評価
- 広域ネットワーク環境での性能評価
  - ordering mix で性能低下
    スループットの低下
    レスポンス時間の増加
  - Pangeaの修正提案
    COMMIT処理修正の提案
  - 性能の改善

今後の課題

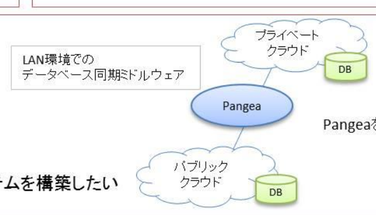
- 実験結果の深い考察
- Pangeaの更なる修正
- レプリカ数を増やした場合

研究目的

ハイブリッドクラウド間のシステム協調が必須
データベースは重要度の高いシステム

⇩

ハイブリッドクラウドにおけるデータベース同期手法の提案
ハイブリッドクラウド上でも性能が落ちることのない新しいシステムを構築したい



Pangeaを介して両者のDB同期
⇒距離による性能低下を予想
⇒Pangeaを修正し性能向上を図る

Android端末におけるThick/Thinクライアント制御機能の実現
（研究担当:本橋　史帆）

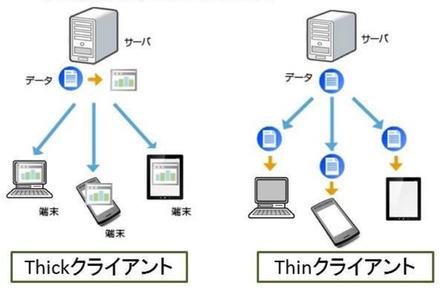
研究背景

サーバ・クライアントモデル

サーバ・クライアントモデルにはThickクライアントとThinクライアントの2種が挙げられる

➢ Thickクライアント
データやアプリケーションを端末上に保持し，処理等もすべて端末上で行うモデル．
⇒オフラインでも操作・使用可能
⇒モバイル端末のThickクライアント化

➢ Thinクライアント
データやアプリケーションをサーバで管理し，ほとんどの処理をサーバで行うモデル．
端末はコマンド入力や画面表示といった薄い役割しか持たない．
⇒サーバ接続のため，ネットワーク環境必須
⇒高機能・高性能処理可能



近年の動向

◆ 無線通信やモバイル通信の高速化
⇒セキュリティ面で注目されているThinクライアント端末の増加
◆ モバイル端末の高機能化
⇒端末上でも高機能・高性能処理が可能に．(端末に依存)

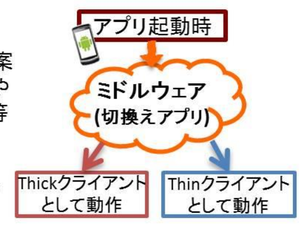
端末	Thickクライアント	Thinクライアント
処理	端末上	サーバ側
セキュリティ	×	○
オフラインでの使用	○	×
高機能/高性能処理	○	◎

研究目的

Thick/Thinクライアントにはメリット・デメリットがある(上表)．
Thickクライアントとして動作するAndroid端末を，環境に応じてThinクライアントに切換えて動作させる制御機能を実装することで，両者のメリットを生かした端末の利用を図る．

提案手法

アプリケーション起動時に，提案ミドルウェアを介し，端末情報やネットワーク環境，アプリ情報等を取得．
得られた情報をもとに，環境に適したモデルに切換えてアプリケーションを起動．



基礎実験

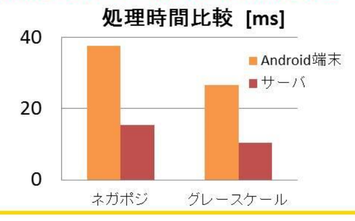
Thickクライアントよりも Thinクライアント向けの処理があり，切換え制御を提案することに意義があることを示す．

実験環境

□ サーバ(CPU:3.60GHz,メモリ4GB)
Ubuntu13.04
JDK (Java Development Kit)
Android SDK
Eclipse
ADT (Android Development Tools)
OpenCV-2.2.0，OpenCV for android-2.4.7
□ クライアント
Galaxy Nexus (Androidバージョン4.2.2)
OpenCV Manager

実験結果

処理時間比較は右下図のグラフの様．
Thinクライアントとしてサーバで処理をした場合にはThickクライアントとして端末上で処理をした場合よりも半分以下の時間で処理できている．
⇒Thinクライアントの処理性能が優れている
⇒Thick/Thinクライアント切換えには意義がある



実装方法

◆ 情報取得
切換え制御に必要な情報として
・端末情報：搭載メモリ容量・使用可能メモリ量等
・ネットワーク情報：通信の有(モバイル通信or Wi-Fi)無
・Wi-Fi接続時：受信信号強度(RSSI)
を取得．

◆ Thickクライアントとして動作させる場合
⇒インテントの利用
➢アプリケーションの中の各機能を橋渡しする仕組み
➢あるアプリを介して他のアプリの起動が可能に

◆ Thinクライアントとして動作させる場合
⇒socket通信の利用
以下の手順でサーバに情報を送り，処理を行う．
1. Android/サーバに同じ処理をするアプリケーションを用意
2. Android側から選択したアプリケーション名をサーバへ送信
3. サーバ側は受信したものと同じ処理をするプログラムを実行
☆Android側ではデスクトップ共有アプリケーション等でサーバ側の処理結果を表示．

まとめ

Thick/Thinクライアント使い分けの意義を証明．
切換えアプリケーションという形で制御機能を提案．
必要となる1つ1つの機能を実装．

http://ogl.is.ocha.ac.jp/

oguchi@computer.org