

小口研究室 研究紹介 (2011年度)

(お茶の水女子大学理学部情報科学科)

無線LAN環境におけるハンドオーバー時の移動端末の特性評価 (研究担当: 森内 彩加)

研究背景

- 無線LANの普及&マルチメディア通信の増加
 - Uplinkのストリーミング増加
- 移動端末の普及

TCP-AV

- QoS-TCPの一つ
- TCP-Renoを拡張
- GreedyなTCP

不公平

不公平の原因

◆同様の通信で端末間のスループットに大差

実験環境

AP1 <802.11g> 送信端末N1台

AP2 <802.11g> 送信端末N2台

HO (Handover)

20m

受信端末

検討課題

- Android端末の場合
- ハンドオーバーでも公平なスループットが確保できるかどうかを調査
- TCPとTCP-AVとの振舞いどのような差が出るかを調査
- 不公平に影響を与える端末数やAPバッファサイズなどの環境を変化
- Androidと従来のノートPCの輻輳ウィンドウを比較評価

Android端末の不公平

ノートPC

Android端末

輻輳が起きてウィンドウが下がると上がらない

タイムアウトを待つと再送

ACKが返ってこない端末は輻輳ウィンドウサイズを指数的に縮小することが出来ない

実験環境

AP1 <802.11g> 送信端末N1台

AP2 <802.11g> 送信端末N2台

HO (Handover)

20m

受信端末

実験結果

◆APのバッファサイズが小さいほどTCP-AVは割り込みやすい

→バッファサイズが小さいほどバッファあふれにより背景端末のTCP-ACKが失われやすいため

◆移動端末がAPIに十分近接し、背景端末数が少ないときのみ割り込める

N1が7台、N2が2台の時

Android端末の不公平

ノートPC

Android端末

輻輳が起きてウィンドウが下がると上がらない

タイムアウトを待つと再送

ACKが返ってこない端末は輻輳ウィンドウサイズを指数的に縮小することが出来ない

実験結果

◆ノートPCを使用した環境で起こっていた不公平がAndroid端末を使用した環境では起こらない

→ノートPCとAndroid端末の輻輳制御に左図のような違いがあるため

Cassandraを用いたKVSデータ処理性能の評価 (研究担当: 菱沼 直子)

研究背景

- データの爆発的な増加
 - 従来のデータ管理システムでは不満が出る場面が生じる
- 新しいデータ管理システム
 - NoSQLが注目される
- NoSQLの実装は未だ発展途中
 - Apache Cassandraと呼ばれる実装に着目

NoSQL...Not Only SQL

RDBMSが得意な分野はRDBMSに RDBMSが不得意な分野にはその分野にあった適切なモデルウェアを使用するという考え方

Apache Cassandra

- Facebook社が開発した分散データベース
- 主な特徴
 - 非中央集中型で単一故障点がない
 - KVS(Key Value Store)型のデータベース
 - カラム型データ構造を持つリッチデータモデル
 - 一貫性の程度を自由に設定可能
 - Cassandra固有の特徴

一貫性レベル

- ユーザが書き込み読み出しにおいて自由に設定可能
- 一貫性の強さを調節可能
- Cassandraの一貫性の強さを判断する方程式

$$R + W > N$$

R: 読み出しレプリカ数
W: 書き込みレプリカ数
N: レプリケーション数

◆この方程式を満たすと強一貫性を達成

実験

◆使用したワークロード

Write Only

Read Only

YCSB Client YCSB-0.1.3

Cassandra on Cloudera cassandra-1.0.6

実験結果

一貫性を考慮した性能測定

- Cassandra側の設定
- レプリカ数: 3
- 一貫性レベル: ONE/TWO/ALL (書き込み読み出しそれぞれ指定)

YCSB側の設定

- レコード数 (1レコード1KB): 100万件
- オペレーション数: 10万件

スレッド数を変化した性能測定

- Cassandra側の設定
- レコード数 (1レコード1KB): 100万件
- オペレーション数: 10万件
- スレッド数: 1~10

実験結果

Latency

write only

read only

一貫性を考慮した性能測定結果

スレッド数を変化した性能測定結果

◆スレッド数が増加すると、スループットも増加

→ノード数が増加しても、スループット性能が向上

◆スレッド数が1~5まではスループットが増加、その後飽和

→一定の値までは負荷を大きくしても効率よく処理可能

緊急災害時に有用な家族間の個人情報共有システムの提案と実装 (研究担当: 長谷川 友香)

研究背景

東日本大震災の発生から... 家族が災害発生時にどこで何をしていたかの情報が役立つ

個人情報共有手法の実現が求められる

システム要件

- 緊急時のみ家族間で情報上の共有を可能にしたい
- 緊急度に合わせて制御を行いたい

閲覧が許可される情報

高 災害発生 すべて閲覧可能

緊急度 連絡がつかない 一部閲覧可能

低 平常 閲覧不可

システム概要

メール、移動履歴、スケジュールといった個人情報をクラウド上に蓄積

クラウド

ユーザA

ユーザB

ユーザC

移動履歴まで互いに閲覧可能

互いに閲覧不可

開発環境

サーバ側: Google App Engine

クライアント側: Android端末

認証レベル

- 3: メールまで閲覧可
- 2: 移動履歴まで閲覧可
- 1: スケジュールまで閲覧可
- 0: 閲覧不可

開発環境

- ユーザごとに認証レベルを保持
- 認証レベルに個人情報を割り当て
- ユーザ間で認証レベルを用いて情報の共有可・不可を決定

階層型相互認証

ユーザA

ユーザB

ユーザC

相手に通知

各ユーザに対して認証レベルを保持

緊急時に相手の認証レベルを変更

- レベルが変更されたユーザにメールで通知
- ユーザ間でレベルが同じなら情報を互いに閲覧可能

システム構成

Google

アカウント

スケジュール

Google App Engine

個人情報共有システム

アカウント情報

移動履歴

参照

参照/変更

メール

アクセス制御部

ユーザA

ユーザB

ユーザC

スケジュールは各ユーザのGoogleカレンダー情報を利用

移動履歴とメールはシステム上に蓄積

システム実装

◆PC用WebサイトとAndroidアプリから情報の閲覧が可能

◆ユーザー画面から他のユーザーの認証レベル変更や情報の閲覧操作が可能

Androidアプリ実行画面

今後の課題

- 外部情報を用いた緊急時判断
- 緊急地震速報や公的機関のサイトの利用 → Officialな判断
- TwitterなどのSNS情報を解析して利用 → Socialな判断
- 緊急時にクラウドやシステムはどうあるべきかの検討