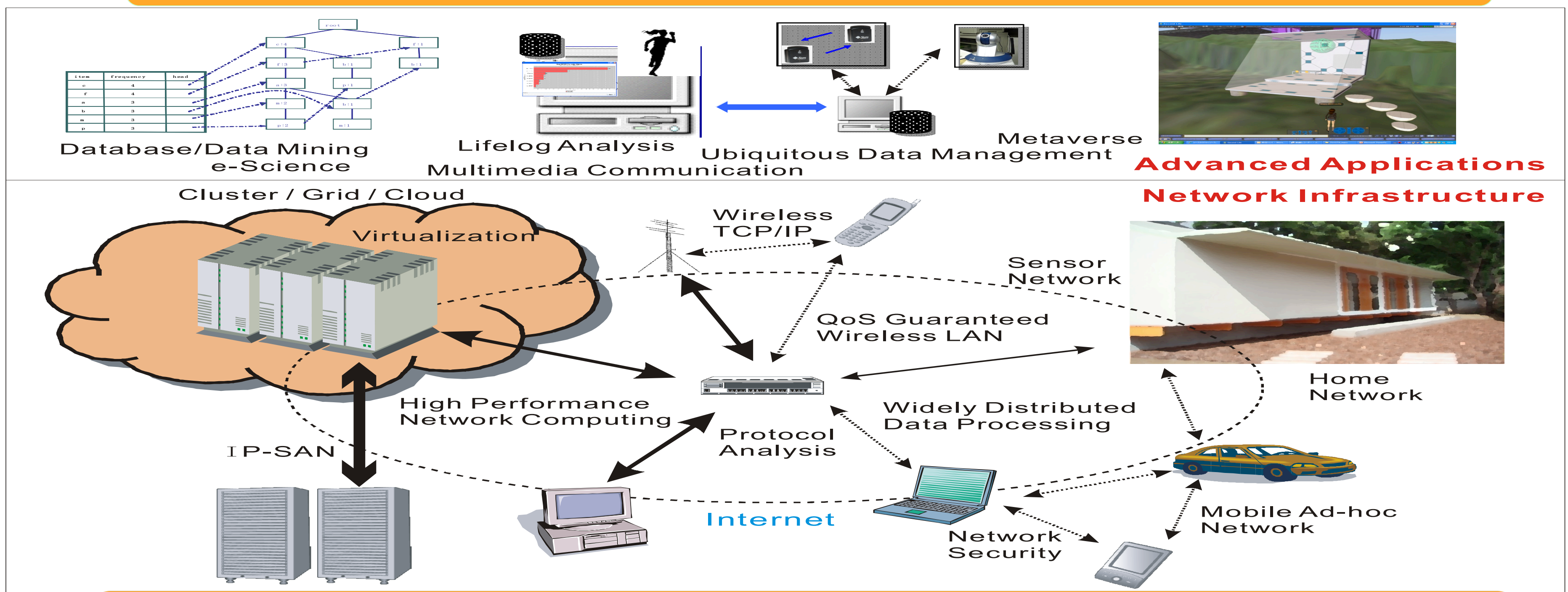


小口研究室 研究紹介 (2020年度)

(お茶の水女子大学理学部情報科学科)

次世代ネットワークコンピューティング基盤と先進的アプリケーション



◆研究テーマ: ネットワークコンピューティング・ミドルウェア

- 多種多様な通信・計算機器が複雑に結びついて情報化社会のシステムを形成
- 次世代ネットワークコンピューティング基盤に焦点を当て、先進的アプリケーションそれを支えるミドルウェアを研究

エッジ、クラウド間分散処理を用いたリアルタイム動画解析 (研究担当: 高崎 智香子)

研究背景

防犯カメラなどの動画データの活用
ディープラーニング技術の応用

- 正確な識別処理には大量のデータ収集・処理が必要
- リアルタイムに識別処理を行うのは非常に困難
- 大量のデータを収集するクラウドにおける急激な負荷上昇

センサとクラウドで識別処理を分散

プライバシーの問題
通信コストが膨大
帯域の圧迫
処理遅延

センサ側における前処理、クラウド側における識別処理の並列化によって負荷分散

OpenPose
姿勢推定ライブラリ
135の特徴点を認識可能
特殊センサを使わずに解析

Flink
耐障害性に優れた分散ストリームフレームワーク
高スループット・低レイテンシを実現

研究概要

各家庭のセンサで収集した動画データを静止画に変換後、OpenPoseを用いて特徴量データに変換。特徴量データのみをクラウドに送信し、機械学習の推論により動作を識別

Sensors at user's home → Cloud → User

Flink: Distributed stream processing
Kafka Consumer, ML Inference, Kafka Producer
Kafka Broker
Analysis Results

実験概要

- Master 1台
- Producer = Brokerへデータ転送
- JobManagerが各TaskSlotにタスクを分配
- Worker 4台
- 各TaskManagerに16台のTaskSlotを配備
- ConsumerがBrokerからデータを購読
- 学習済みモデルを用いた推論で動作識別

実験結果

提案システムの解析スループット、総遅延時間を測定

- 解析なし
- NN解析
- LSTM解析

データ転送レート > 処理レートにより、タスクが停滞
実験開始から20%のタスクの平均総遅延時間で比較
1データの推論時間
NN: 平均0.508ms, LSTM: 平均1.835ms

解析なし: スループット向上せず
Producerからのデータ転送、並列化のオーバーヘッド
NN・LSTM解析: スケーラブルな解析
並列化によりスループット向上、各タスクの遅延削減

全解析パターンで(c)の設定により解析性能改善
多少のオーバーヘッドがあっても、逐一データ転送することで、各タスクの遅延を削減可能

まとめと今後の課題

まとめ
エッジ・クラウド間分散動画解析システムを構築
機械学習の推論のような高負荷な解析をスケラブルに処理可能

今後の課題
エッジ・クラウド間のネットワーク遅延を考慮
家庭で利用されるエッジデバイスを用いたより実環境に近い環境での評価

大規模環境における動的実行領域分割ジョブスケジューリング手法の提案と評価 (研究担当: 高山 沙也加)

研究背景

- 近年のHPCシステムでは投入されるジョブ数は増加しており、要求条件も多様化
 - システム規模は増加傾向
- 運用者とユーザーでは重視する点は異なる
 - 運用者: 電力、充填率
 - ユーザー: 待ち時間
- 適用するスケジューリングアルゴリズムによって充填率や待ち時間は大きく変わる

重視: 待ち時間 (ユーザー)

重視: 電力 充填率 (運用者)

効率的な運用で性能の向上が望める

ユーザー: ジョブ投入、ジョブ数決定

運用者: ノード分割、ポリシー決定、システム監視

スケジューラ

HPCシステム

システムノード

提案手法

- 過去の投入ジョブ分布とジョブの実行時間、そして運用者が設定したジョブの最大最小待ち時間の差から、自動で実行領域の分割条件を決定するアルゴリズムの提案
- 大規模ジョブの待ち時間の削減によってジョブ規模ごとの待ち時間のばらつきを小さくする
- 決定する分割条件
 - 実行領域の分割数: ユーザーの指定の最大最小待ち時間の差に応じて変更
 - ジョブの必要ノード数範囲: ジョブ粒度が揃うように調整
 - 各実行領域のサイズ: ノード実行時間積を割合にして全体ノード数に合わせて分割

Node of Zone A (N_A)

Node of Zone B (N_B)

Node of Zone C (N_C)

(t) 時間

実験・分析結果

- Slurm Workload Manager オープンソースのジョブスケジューラ
実行ユーザーに対してリソースへの排他的・非排他的なアクセス割り当て
- 利用ジョブミックス
「京」分布: 「京」ジョブ統計情報から生成
混合分布: 「京」分布の小規模ジョブ割合を増加
- 充填率
 - 「京」分布では分割数を増やしても充填率の悪化はほぼみられなかった。
 - 混合分布では分割数を増やすほど充填率が悪化した。これは投入ジョブに偏りが生じやすくなるため。
- 待ち時間
 - 「京」分布では基本的には分割なしより改善したがjob2のみ分割数を増やすと悪化した。
 - 混合分布では4分割以上は分割数を増やすほど悪化した。こちらも投入ジョブの偏りのため。

最小ノードサイズをノード実行時間積に応じて可変にすることで、著しく待ち時間の長いジョブを大幅に減らせた。

今後の課題

- 特定の実行領域に偏りが生じた場合の対策
 - 一定時間隔ではなく、ジョブ投入の偏りを検知して
 - 複数ジョブでコンテナを作り他の実行領域に投入