

組み込み機器による大規模計算環境の開発

共生アプリケーション

- ユーザの活動を適応的に支援
- 能力を最大限発揮できるようサポート
例)
 - どこでも外国語の語彙学習
 - 脳の活動に応じて気分合う音楽を自動選択
 - 話者を認識して選択的にノイズ・キャンセリングなどなど

現状と方針

- 現状の計算基盤
 - 複雑なアプリケーションに対しては、携帯/サーバともに性能がまだ不十分
 - 消費電力が大きく、筐体もまだ大きい
- 方針
 - 携帯/サーバのさらなる性能改善
 - それぞれのさらなる省電力化
 - いくつかのサブテーマにわけ、集中的に取り組む

共生情報システム

- アプリケーションは携帯情報端末で動作
- 場合によってはサーバ上のデータを活用



サブテーマ

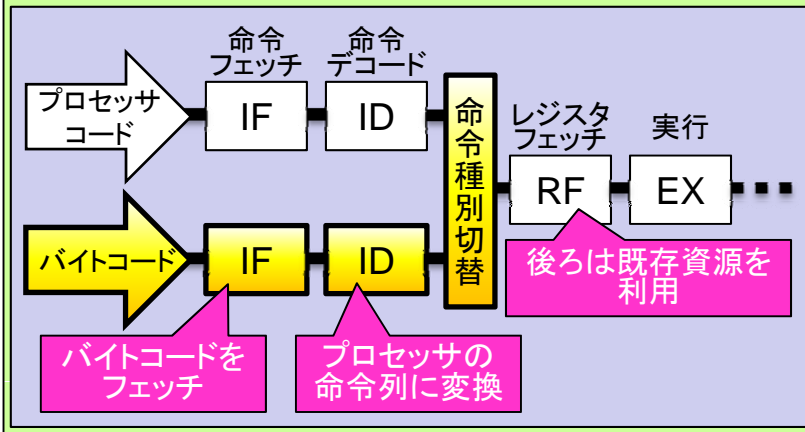
- 携帯側
 - Dalvik アクセラレータ: Android 携帯向け Java アクセラレータ
- サーバ側
 - Scheduled Trace Cache を用いた高速な命令発行
 - 小容量 CAM を用いたマップ表の回路面積削減手法
 - キャッシュ・ヒット率改善のためのデータ配置手法
 - GPGPU による Hilbert-Huang 変換の高速化



サブテーマ紹介(1)

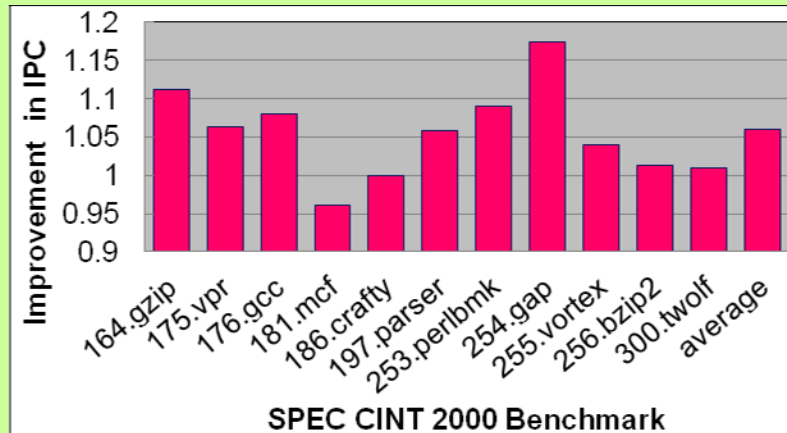
Dalvik アクセラレータ

- 携帯電話向けプラットフォーム Android 上のアプリケーション実行環境 Dalvik VM の実行を高速化
- アプローチ
 - Dalvik バイトコードをプロセッサが直接解釈
 - バイトコードのデコーダ等を追加
 - プロセッサネイティブな命令列に動的に変換して実行
 - プロセッサシミュレータに搭載して評価する
- 現状
 - Dalvik VM ならびにベンチマークソフトの稼働



Scheduled Trace Cache

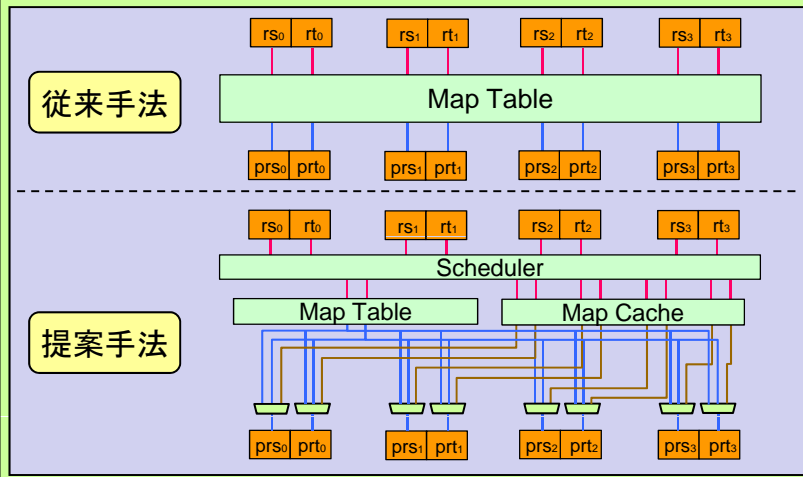
- OoO スーパースカラ・プロセッサのフロントエンドは複雑
 - ➡ 分岐予測ミスペナルティの増加
- アプローチ
 - Scheduled Trace Cache に発行済みの命令を保持
 - 命令はリネーミングもスケジューリングも終えた状態
 - ヒットした場合には単純な処理(フェッチ+分解)で命令を発行
- 平均 6.01%, 最大 17.4% 性能を改善



サブテーマ紹介(2)

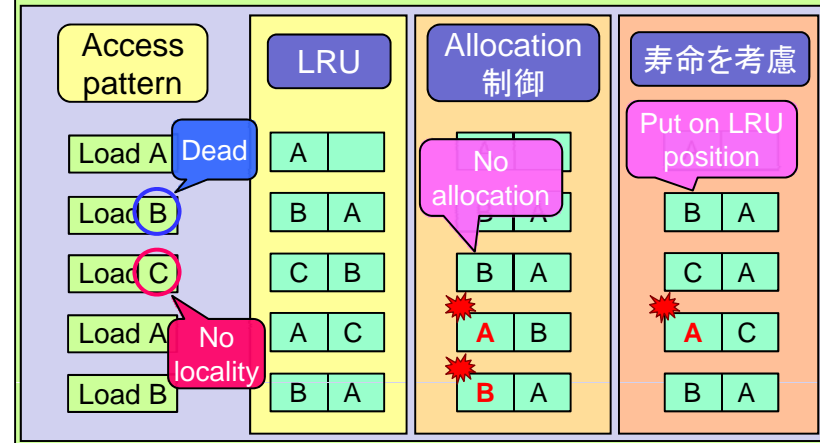
マップ表の回路削減

- レジスタ・マップ表
 - 12ポート, 32エントリ程度のRAM
 - 回路面積が大きい → 消費電力も大きい
 $(RAM \text{ の面積}) \propto (\text{エントリ数}) \times (\text{ポート数})^2$
- アプローチ
 - レジスタ・マップ・キャッシュ
 - 12ポート, 8エントリ程度のCAM
 - キャッシュにミスする場合にのみマップ表を参照
 - マップ表のポートは少なくてよい



キャッシュのデータ配置

- 近年のプロセッサではキャッシュが重要な役割を担う
- 一般的にはLRUでリプレース
 - 意外と無駄が多い
 - 使われないデータがキャッシュに残ることも
 - A) 局所性のないデータ (例: ストリーム・データ)
 - B) 寿命が尽きたデータ
- アプローチ
 - A) 局所性がないデータをキャッシュに置かない
 - B) 寿命が尽きたデータを追い出しの候補へ

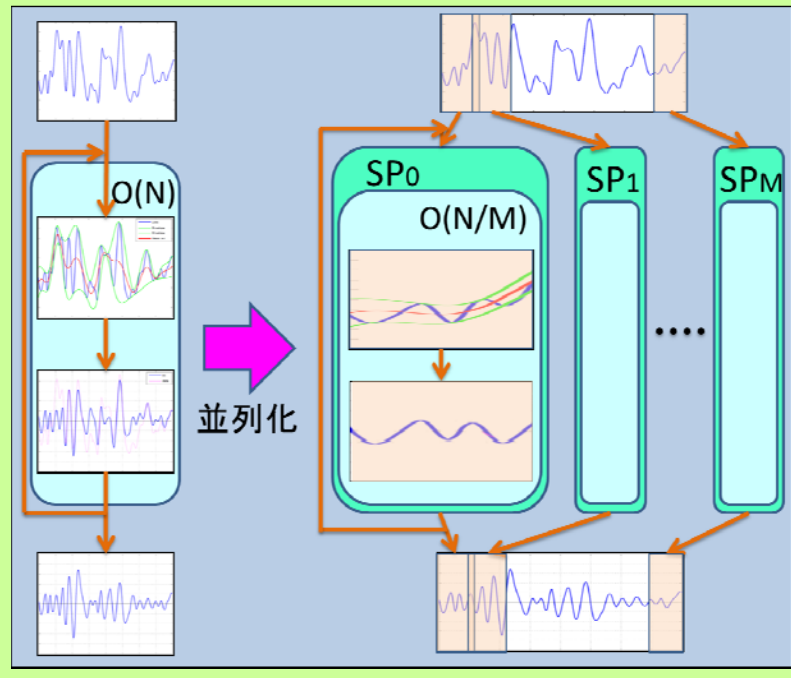


組み込み機器による大規模計算環境の開発

サブテーマ紹介(3)と今後の展望

GPGPU

- 近年GPGPUによる汎用計算の高速化が注目
- Hilbert-Huang 変換の高速化
 - 経験的モード分解(EMD) + Hilbert 変換
- EMDの並列化アプローチ:



本年度の成果

- 携帯側
 - プロセッサ・シミュレータ上で Android と Dalvik VM の稼働に成功
- サーバ側
 - Scheduled Trace Cache の実装と評価
 - マップ表の回路面積削減手法の実装
 - キャッシュのデータ配置手法の実装
 - 並列 EMD を実装

今後の展望

