Making Cellphones Fly: First Experiences from the Trenches

{roel.wuyts/michel.tilman}@imec.be

BENEVOL2006, Delft
December 12th, 2006
IMEC Context: Mobile systems

- Wide variety of devices and applications
  - Ubiquitous computing (third wave of ‘calm’ computing)

- Requirements
  - 100 Applications / 80 platforms
  - 100000000 LOC
  - Video CODEC, 3D engines, wireless

- Programming languages
  - C, MATLAB, C++, Java

- Runtime
  - Often no OS
  - Heterogeneous platforms

- Cost
  - Resources, energy, speed
Examples

• **Power management**
  - Storage and transfer of data
    • Retrieving addition operands from main memory and storing the result consumes 100x more energy than the actual addition
      - MPEG-4 decoding, 3D graphics

        ```
        for (int i = 0; i < MAX_ROWS; i++) {
            for (int j = 0; j < MAX_COLUMNS; j++) {
                A[i][j] = calculate(A[i][j]);
            }
        }
        ```
    - Networking

• **Processor frequency**
  - Non-linear correlation
  - Parallel processing
    » Up to 100 heterogeneous cores in 2012
IMEC Methodology

- **High-level description of platform**
  - Number of ports on DMA controller, memory hierarchy, access times, number of processors

- **Analyze code**
  - Static
    - Global!

    ```java
    switch (shape.type) {
    case circle: calculateCircleArea(shape);
    case rectangle: calculateRectangleArea(shape);
    case triangle: calculateTriangleArea(shape);
    }
    ```

  - Dynamic (profiling)

- **Transform code**

- **Compile**
Challenges

- Analyzing C / C++ code is hard
  - Fully automated optimization process impossible in general
  - Mixed mode tool chain
    - Different input languages
- Applications increasingly dynamic
  - Memory allocations
  - Polymorphism
  - Software configuration
  - Language, middleware, explicit parallelism?
- Global analysis assumes total control
  - Maverick
- Self-tuning and adaptive runtime systems
- Hybrid approaches
From the Trenches into Battle...

- **Finding a decent C/C++ Parser**
  - iPlasma, Antlr, StrategoXT, Columbus, EDG, GCC, ...
  - Settled for the CDT DOM Parser

- **Constructing an IDE**
  - Settled for Eclipse

- **Status: Read&Write C code**

- **Next up: domain-specific analysis&refactoring**
  - call-graph
  - data-flow graph
  - Declarative Meta Programming
  - bridging to existing IMEC tools