THE EMBEDDED JAVA BENCHMARK SUITE

Matin Schoeberl (DTU), Thomas P. Preusser (TU Dresden), Sascha Uhrig (U. of Augsburg)
Benchmark targets and related work

Benchmark categories

Some numbers

Open questions

Summary
Benchmark Targets

- Real applications
- WCET analyzable
- Embedded Java (J2ME)
  - CLDC 1.1 library - tested with Sun Squawk
  - SCJ JDK subset
- Multi-processor scaling measurements
RELATED WORK

- CaffeinMark
  - Kernel benchmarks
- CDx
  - RTSJ, single periodic thread
- PapaBench
  - Will be presented tomorrow
GENERAL PROPERTIES

- Big variation of embedded systems performance
- Runtime should be reasonable
- Self adapting
  - Increase iterations till one second elapsed
  - Result is iterations per second
- Micro benchmarks
- Kernels
- Applications
- Multithreaded
Micro Benchmarks

- Measure single / two bytecodes
- Useful for Java processors and interpreting JVMs
- Two measurement loops
  - Subtract overhead loop

Friday, 20 August, 2010
Micro Measurement

public int perform(int cnt) {
    int a = 0;
    int b = 123;
    int i;
    for (i=0; i<cnt; ++i) {
        a = a+b+b;
    }
    return a;
}

public int overhead(int cnt) {
    int a = 0;
    int b = 123;
    int i;
    for (i=0; i<cnt; ++i) {
        a = a+b;
    }
    return a;
}
Embedded Java applications in industrial use

- Kfl, Lift, UdpIp
- WCET analyzable
- Developed for JOP
- Looking for more *external* applications
WCET ANALYSIS

- Provide loop bounds
- Use only analyzable libraries
  - No hash tables, ...
- Application benchmarks are analyzable
  - Used to test WCET tools
# Application Results

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Squawk/MacBook 2.5 GHz</th>
<th>picoJava II 40 MHz</th>
<th>JOP 100 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kfl</td>
<td>121814</td>
<td>23322</td>
<td>24058</td>
</tr>
<tr>
<td>Lift</td>
<td>149114</td>
<td>25244</td>
<td>24308</td>
</tr>
<tr>
<td>UdpIp</td>
<td>61478</td>
<td>11736</td>
<td>10144</td>
</tr>
</tbody>
</table>
• Embedded Java goes CMP
  • Hope for more performance
• Scaling tests
  • Simple scaling examples (automatic scaling)
• Pipeline application
• General multithreaded application missing
MT BENCHMARKS

- Guide CMP development
  - Memory subsystem (caching)
  - Memory controller
  - Locks
- Alternatives to locks - transactional memory
Speedup Example

- 8 vs. 1 JOP cores

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix multiplication</td>
<td>6.5</td>
</tr>
<tr>
<td>NQueens</td>
<td>6.5</td>
</tr>
<tr>
<td>Raytrace (6 threads)</td>
<td>4.2</td>
</tr>
<tr>
<td>(Lift)</td>
<td>6.6</td>
</tr>
</tbody>
</table>
FUTURE WORK

- More applications
- SCJ wrappers
- Real-time measurements
OPEN QUESTIONS

- What are real-time benchmarks?
- Low-level interrupt latency, scheduling overhead?
- Reduce periods till deadline miss?
- Measure slack time and/or jitter?
Benchmarks for embedded Java
Some real world applications
Open-source at SF
Wiki for collection of results