Automated Refactoring for Loop Parallelism in Java

Cosmin Radoi, “Politehnica” University of Timisoara
ADVISORS: Danny Dig, Ralph Johnson

Motivation

• The hardware multi-core revolution is here
• Existing software is designed for single-core and needs to be rewritten
• Rewriting software manually is labor-intensive and error-prone
• Java provides the ParallelArray library that executes array operations (e.g., map, apply, reduce) in parallel

Goal

Implement an interactive Eclipse plugin that enables Java programmers to parallelize loops using ParallelArray
• programmer selects an array
• tool analyzes the safety of the transformation and rewrites the code

Challenges

• Correctness: tool analyzes whether the loop iterations can be executed in parallel
• Performance: the analysis and rewriting should be fast - our aim is under 30s

Refactoring Flow

Original source
Find array accesses
Array element type mutable?
Yes
Instrument code for WALA
No
Determine parallel operator
Determine non-aliasing
Rewrite code
Refactored source

Preliminary Results

<table>
<thead>
<tr>
<th>Project</th>
<th>Created operators</th>
<th>Other changes</th>
<th>Changed LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBJ Parts of Speech Tagger</td>
<td>1</td>
<td>4</td>
<td>26</td>
</tr>
<tr>
<td>LBJ Coreference</td>
<td>1</td>
<td>9</td>
<td>17</td>
</tr>
<tr>
<td>Jolden Barnes &amp; Hut N-body</td>
<td>4</td>
<td>10</td>
<td>83</td>
</tr>
<tr>
<td>Jolden Em3D</td>
<td>2</td>
<td>8</td>
<td>23</td>
</tr>
</tbody>
</table>

Examples

Init

```
Product[] a = new Product[length];
```

Apply

```
for (int i = 0; i < a.length; i++)
    a[i].price += 1;
```

```
for (int i = 0; i < a.length; i++)
    if (a[i].price > 100)
        a[i] = new Product(a[i].price - 5);
```

Map

```
a.apply(new Procedure<Product>() {
    public void op(Product elt) {
        elt.price += 1;
    }});
```

```
a.replaceWithMapping(new Op<Product,Product>() {
    public Product op(Product elt) {
        if(elt.price > 100)
            elt = new Product(elt.price - 5);
        return elt;
    }});
```

Reduce

```
for (int i = 0; i < a.length; i++)
    total =
        new Product(total.price + a[i].price);
```

```
a.reduce(new Reducer<Product>() {
    public Product op(Product total, Product elt) {
        return new Product(total.price + elt.price);
    }});
```

Information Trust Institute
University of Illinois at Urbana-Champaign
www.iti.illinois.edu