DeCoP: Deterministic Cooperative Parallelism

Section Based Thread Escape Analysis

Jose Renau & Cormac Flanagan

University of California, Santa Cruz
http://masc.soe.ucsc.edu
Problem

Multithreading is too difficult

- Multithreading introduces several problems:
  - …requires synchronization directives
  - …memory consistency models
  - …may not even be deterministic
  - …harder debugging
  - …
Possible Mitigating Solution

Use simpler cooperative model

• In cooperative multithreading
  • …one thread executes at a given time
  • …context switches only when the programmer indicates it
  • …cooperative model is used for handling IO
  • …but lacks parallelism (thing to address in the proposal)
This Talk approach to get it working

Improve Thread Escape Analysis

• If we can prove that an access is local
  • ...no need to do versioning in TM
  • ...much faster to do tests for debugging
  • ...
Dynamic Loads Proved no Sharing
What is the main Trick??

Detect Thread Sections

Thread 1

<table>
<thead>
<tr>
<th>Thread 1</th>
<th>Thread 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>W_x CS1</td>
<td>W_x CS1</td>
</tr>
<tr>
<td>R_z CS2</td>
<td>R_z CS3</td>
</tr>
<tr>
<td>R_x CS2</td>
<td>W_y CS3</td>
</tr>
<tr>
<td>R_y CS3</td>
<td>W_z CS5</td>
</tr>
</tbody>
</table>

Program Flow →
Possible Use: TSAN

Key points:

- Thread Sanitizer increased program execution time 12.5 times
- Even with no instrumentation, execution time was 6.5 times native
- SBPA improved runtime by 2.74 times
- Base improved by 1.116 times
Current Directions

- Leverage Javascript cooperative programming model
  - Add parallelism without breaking model and minimal effort

- Prove cooperative programs ran in parallel as data race free
Questions?

Jose Renau & Cormac Flanagan

University of California, Santa Cruz
http://masc.soe.ucsc.edu