Parametric Analysis for Adaptive Computation Offloading

Presenter: Hao Li

University of Minnesota

lixx4002@umn.edu

February 17, 2015
Overview

- Motivation
- Min-cut Model
- Issues and Solutions
- Cost Factors and Algorithm
- Performance
- Summary
Optimal code generation is sensitive to execution instances such as execution options, input parameters and data files.

Optimal program partitioning for computation offloading depends on tradeoff between the computation cost and communication cost.
Min-Cut Model

audio encoding application

```c
f() {
    for (j = 0; j < x; j++) {
        f1: for (i = 0; i < y; i++) {
            get audio sample p;
            inbuf[i] = p;
        }
    }
    g();
    f2: for (i = 0; i < y; i++) {
        q = outbuf[i];
        write q to output;
    }
}
```
Question: Why does not the paper before use this method to get optimal solution?
Issues

- Create TCFG
- Redundant data transfer
- Inexact data dependency information
- Program partitioning constraints
Create TCFG

- Task: statement segment that starts with task header, ends with task branch
- No task branches in the middle
Create TCFG

Why not to use function as nodes?
Redundant data transfer

\[ V_{si}(v, d) : \text{Is data item } d \text{ valid on the server at the entry of task } v? \]

\[ V_{so}(v, d) : \text{Is data item } d \text{ valid on the server at the exit of task } v? \]

\[ V_{ci}(v, d) : \text{Is data item } d \text{ valid on the client at the entry of task } v? \]

\[ V_{co}(v, d) : \text{Is data item } d \text{ valid on the client at the exit of task } v? \]
Goal: Honor all data dependency

Pointer analysis techniques: find data bound

Dynamic bookkeeping
  - Registration table
  - Mapping table
  - Translation mechanism
Cost Factors

**Computation Cost**
\[ \sum_{v \in V} M(v)c_s(v) + \neg M(v)c_c(v) \]

**Data Communication Cost**
\[ \sum_{(v_i,v_j) \in E,d} \neg V_{so}(v_i,d)V_{si}(v_j,d)c_{sd}(v_i,v_j,d) + \neg V_{co}(v_j,d)V_{ci}(v_i,d)c_{sd}(v_i,v_j,d) \]

**Task Scheduling Cost**
\[ \sum_{(v_i,v_j) \in E} \neg M(v_i)M(v_j)c_{st}(v_i,v_j) + \neg M(v_j)M(v_i)c_{st}(v_i,v_j) \]

**Data Registration Cost**
\[ \sum_d N_c(d)N_s(d)c_a(d) \]
1: \( \mathcal{Z} = \emptyset \)
2: while \( (\mathcal{X} \neq \emptyset) \) {
3:     Choose a sample parameter value \( \tilde{\lambda} \in \mathcal{X} \);
4:     Find a minimum cut \( P \) for problem 2 with parameter value \( \tilde{\lambda} \);
5:     With \( P \), compute set \( \mathcal{H} \) defined by (7);
6:     Add \( (P, \mathcal{H}) \) to \( \mathcal{Z} \);
7:     \( \mathcal{X} = \mathcal{X} - \mathcal{H} \);
8: }
Question: Difference from CloneCloud
### Performance

**Presenter:** Hao Li (UMN)

#### Program Name & Description

<table>
<thead>
<tr>
<th>Program Name</th>
<th>Description</th>
<th>No. of Parameters</th>
<th>No. of Source Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>rawcaudio</td>
<td>ADPCM in Mediabench, Speech Compression</td>
<td>1</td>
<td>205</td>
</tr>
<tr>
<td>rawdaudio</td>
<td>ADPCM in Mediabench, Speech Decompression</td>
<td>1</td>
<td>178</td>
</tr>
<tr>
<td>encode</td>
<td>G.721 in Mediabench, CCITT Voice Compression</td>
<td>4</td>
<td>1118</td>
</tr>
<tr>
<td>decode</td>
<td>G.721 in Mediabench, CCITT Voice Decompression</td>
<td>4</td>
<td>1248</td>
</tr>
<tr>
<td>fft</td>
<td>FFT in Mibench, Discrete Fast Fourier Transforms</td>
<td>3</td>
<td>332</td>
</tr>
<tr>
<td>susan</td>
<td>susan in Mibench, Photo Processing</td>
<td>12</td>
<td>2122</td>
</tr>
</tbody>
</table>

#### Program Details

<table>
<thead>
<tr>
<th>Program</th>
<th>No. of Tasks</th>
<th>No. of Annotations</th>
<th>No. of Partitioning Choices</th>
<th>Analysis Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>rawcaudio</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>164</td>
</tr>
<tr>
<td>rawdaudio</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>185</td>
</tr>
<tr>
<td>encode</td>
<td>107</td>
<td>4</td>
<td>4</td>
<td>2247</td>
</tr>
<tr>
<td>decode</td>
<td>87</td>
<td>4</td>
<td>4</td>
<td>2159</td>
</tr>
<tr>
<td>fft</td>
<td>26</td>
<td>3</td>
<td>2</td>
<td>748</td>
</tr>
<tr>
<td>susan</td>
<td>95</td>
<td>13</td>
<td>3</td>
<td>3482</td>
</tr>
</tbody>
</table>
Summary

- **Pros**
  - Create a wonderful model
  - Use message to wake up host
  - Take execution instances into account

- **Cons**
  - Analysis Time is too long
  - Network may change dramatically
  - Server tasks and client tasks do not run simultaneously
Questions?
The End