A Memory Efficient and Modular Approach for String Matching

Hoang Le, Edward Yang, Viktor Prasanna
Ming Hsieh Department of Electrical Engineering - University of Southern California
{hoangle,yeyang,prasanna}@usc.edu

Background
- Multi-pattern, large-scale string matching
- Return all occurrences, if any, of every pattern in the given input string at any position.
- Memory efficiency (in bytes/char)
- Throughput (in Gbps)

Contributions
A novel architecture for large-scale string matching:
- Simple architecture
- High memory efficiency (~ 1 B/char)
- High throughput (3.2 Gbps)
- Modularity

Our Approach – Binary Search Tree
- Height of a complete BST bounded by $\left\lceil \log N \right\rceil$ (N - number of nodes)
- Amount of memory doubled in the next level
- No need to explicitly store the addresses of child nodes
- Last few stages moved onto external SRAMs

Data Structure
- Overlapping resolved by merging prefix-patterns with parent patterns ("an", "and" merged into "andy")
- Bitmap vector keeps prefix-patterns, a "1" at position i shows a prefix of length i (ex. "andy_0111")
- Each node consists of a parent-prefix and its bitmap vector
- Standard BST traversal:
  - Start from root
  - Go left if less than or equal, otherwise right

Modularity
- Module used as building block to process long patterns
  - Long patterns are cut into segments of length L
  - Matched label and Matching Status Vector carried to the next module to ensure valid long patterns
- Variable module width (L)
  - Duplicated to improve throughput

Experimental Results

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Memory Efficiency (B/char)</th>
<th>Throughput (Gbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our work / Rogets</td>
<td>1.07</td>
<td>3.2</td>
</tr>
<tr>
<td>Our work / Snort</td>
<td>1.05</td>
<td>3.2</td>
</tr>
<tr>
<td>Field-Merged / Rogets</td>
<td>6.33</td>
<td>1.14</td>
</tr>
<tr>
<td>Field-Merged / Snort</td>
<td>2.16</td>
<td>1.14</td>
</tr>
<tr>
<td>Bit-Split / Rogets</td>
<td>34.1</td>
<td>1.76</td>
</tr>
<tr>
<td>Bit-Split / Snort</td>
<td>28.5</td>
<td>1.76</td>
</tr>
<tr>
<td>Variable-Stride / Snort</td>
<td>2.4</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Rogets dictionary 178K, maximum length = 24
Snort dictionary 146K, maximum length = 232