In SIGCOMM 2008:

Deflating the Big Bang:
Fast and Scalable Deep Packet Inspection with Extended Finite Automata

Randy Smith, Cristian Estan, Somesh Jha, Shijin Kong
Computer Sciences Department
University of Wisconsin-Madison

Presented by Debbie Perouli (depe@purdue.edu)
Purdue CS 636
April 2009
Overview

• **Context**
  o Signatures use regular expressions
  o DFAs: fast and readily combined, but state space explosion
  o NFAs: small, but slow matching procedure

• **Goal:** make deep packet inspection practical at high speeds

• **Contributions**
  i. Formal characterization of DFA state space explosion
  ii. XFA deterministic model (building on previous work)
  iii. XFA systematic optimizations
  iv. Experimental comparison proving XFAs faster and smaller
  v. Hardware architecture for efficient execution of XFAs
State Explosion (BIG BANG!) in DFA Combination

/ab.*cd/ + /ef.*gh/

Theorems:
D1, D2 unambiguous

⇒
D1 + D2 unambiguous and

|D1 + D2| < |D1| + |D2|
\[
X = (Q, V, \Sigma, \delta, U, (q, v), F)
\]

SETS:  
- \(Q\) (states)  
- \(V\) (variable values)  
- \(\Sigma\) (inputs)  
- \(F \subseteq Q \times V\) (accepting states)

FUNCTIONS:  
- \(\delta : Q \times \Sigma \to Q\) (transition)  
- \(U : Q \times V \to V\) (update)
Optimizations

1. Exploiting Runtime Information
   Initialization and Reset of the counter →
   Insert into and Remove from the payload offset list

2. Combining Independent Variables
   - Reduces both state size & instruction counts
   - Similar to register allocation problem

3. Code Motion and Instruction Merging
   Move bit assignment instructions, so that
   those belonging to the same word are adjacent.
Dataflow & Compatibility Analysis

- $f_{\text{set}}(C)$ → Active
- $f_{\text{reset}}(C)$ → Inactive
- $f_{\text{decr-\ and- test}}(C) \rightarrow C$
- $f_{\text{preserve}}(C) \rightarrow C$

<table>
<thead>
<tr>
<th>Inactive</th>
<th>Active</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r,d,p$ set</td>
<td>$r,d,p$ set</td>
</tr>
<tr>
<td>reset set</td>
<td>set set</td>
</tr>
<tr>
<td>reset set</td>
<td>reset set</td>
</tr>
<tr>
<td>decr reset</td>
<td>set set</td>
</tr>
<tr>
<td>pres pres</td>
<td>pres pres</td>
</tr>
</tbody>
</table>

Diagram:

- [ctrSET 1,200, [ALT 3]]
- [ctrDEC 2]
- [ctrRST 1]
- [ctrRST 2]
- [ctrSET 1,150, [ALT 7]]
- [ctrSET 1,2,150, [ALT 7]]
# Combined Automata for several Protocols before & after Optimization

<table>
<thead>
<tr>
<th>Rule set</th>
<th>Num Sigs</th>
<th># States</th>
<th>Variables</th>
<th>Instrs per state</th>
<th>Aux memory (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>DFA</td>
<td>XFA</td>
<td>max</td>
<td>program</td>
</tr>
<tr>
<td>Snort FTP</td>
<td>72</td>
<td>&gt;3.1M</td>
<td>769</td>
<td>50</td>
<td>93</td>
</tr>
<tr>
<td>optimized</td>
<td></td>
<td></td>
<td></td>
<td>38.67</td>
<td>1336K</td>
</tr>
<tr>
<td>Snort SMTP</td>
<td>56</td>
<td>&gt;3.1M</td>
<td>2,415</td>
<td>37</td>
<td>64</td>
</tr>
<tr>
<td>optimized</td>
<td></td>
<td></td>
<td></td>
<td>21.48</td>
<td>2211K</td>
</tr>
<tr>
<td>Snort HTTP</td>
<td>863</td>
<td>&gt;3.1M</td>
<td>15,266</td>
<td>31</td>
<td>52</td>
</tr>
<tr>
<td>optimized</td>
<td></td>
<td></td>
<td></td>
<td>15.91</td>
<td>7445K</td>
</tr>
<tr>
<td>Cisco FTP</td>
<td>38</td>
<td>&gt;3.1M</td>
<td>527</td>
<td>19</td>
<td>26</td>
</tr>
<tr>
<td>optimized</td>
<td></td>
<td></td>
<td></td>
<td>12.35</td>
<td>271K</td>
</tr>
<tr>
<td>Cisco SMTP</td>
<td>102</td>
<td>&gt;3.1M</td>
<td>3,879</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>optimized</td>
<td></td>
<td></td>
<td></td>
<td>5.20</td>
<td>806K</td>
</tr>
<tr>
<td>Cisco HTTP</td>
<td>551</td>
<td>&gt;3.1M</td>
<td>11,982</td>
<td>17</td>
<td>22</td>
</tr>
<tr>
<td>optimized</td>
<td></td>
<td></td>
<td></td>
<td>10.48</td>
<td>4907K</td>
</tr>
</tbody>
</table>

![Histogram of states per DFA and XFA for Snort HTTP](image1)

![Histogram of instructions per state for Snort HTTP](image2)
Memory Usage & Performance

Snort FTP

Snort SMTP

Snort HTTP

Cisco FTP

Cisco SMTP

Cisco HTTP