On the Impact of Caching on high Performance Packet Classifiers

Harald Widiger, Andreas Tockhorn, Dirk Timmermann
University of Rostock
Institute of Applied Microelectronics
and Computer Engineering
Outline

- Classification Problem
- Hash-based Packet Classification
- Cache Architecture
- Performance
- Conclusion

09.12.2008

University of Rostock
Classification Problem

<table>
<thead>
<tr>
<th>Key</th>
<th>Information (Rule)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>Information (Rule)</td>
</tr>
<tr>
<td>Key</td>
<td>Information (Rule)</td>
</tr>
</tbody>
</table>

Memory Access

Mapping Function (Memory Search)

*Frame In* -> Header Parser -> Functional Element -> *Frame Out*
Adaptive Hashing in Hardware

- Collision: \( X \neq Y \); \( H(X) = H(Y) \)
- Resolution
  - Rehashing \( H(H(X)) \);
  - Linear \( H(X) + \text{Prime} \)
- Time Complexity: \( O(1) \)
- Memory Space: \( O(N) \)
An Evolvable Hash Function

Genome: \( M \cdot 3 \cdot \log_2(N) \)
Genetic Algorithm – Implementation in Hardware

- **μ** Individuals
- **λ** Offspring
- Mutation Rate: \(2/\lambda\)
- Survivor Selection: \(\mu\) new parents out of \(\lambda\) offspring and fittest old parent; \((\lambda, \mu)\)-elitist
Hash-based Packet Classification - EPC

- Two parallel (data) paths
  - Packet classification
  - Hash function evolution
- Classification and Evolution work in parallel
- When finding a better hash function, data path is switched to the better one
Collision Distribution

- 32768 keys
- Up to 22 collisions
- Number of keys with many collisions very small
Caching in Packet Classifiers

- Number of collisions = memory accesses → Cache the entries with most collisions
- Cache: memory accesses = 1 → do not cache entries that occur most often
Cache Architecture

- Implemented in parallel to the data path
- Size and degree of associativity configurable
- for each degree one BRAM and a comparator
- Constraint: size/associativity = $2^N$
Cache Size vs. Performance

32768 keys

Good results already when caching 0.78% of keys
Associativity - Costs

![Diagram showing costs associated with associativity. The x-axis represents associativity levels (1, 2, 4, 8), and the y-axis shows logic resources in slices and BRAMs.]
Associativity - Performance

![Graph showing the relationship between associativity and performance metrics.](image)

- **Fitness difference [%]**
  - Fitness without Cache
  - Fitness with Cache
  - Fitness with optimal Caching
  - Difference to Optimum [%]

**Axes:**
- **X-axis:** Associativity (1, 2, 4, 8)
- **Y-axis:** Collisions (0 to 12,000)

**Key Points:**
- The graph illustrates the performance impact of varying associativity levels.
- Comparisons are made between different cache configurations and their fitness differences.

**Date:** 09.12.2008

**Institution:** University of Rostock
With cache the max. number of memory accesses is largely constant
Conclusion

- When using Hashing a cache should be used to limit the max. number of memory accesses
- Even small caches are effective
- Higher associativity improves effectiveness gradually
- Even with larger keysets, it can be expected to have the max. number of memory accesses kept widely constant