Layered Interval Codes for TCAM-based Classification

Anat Bremler-Barr
Boris Farber
The Interdisciplinary Center, Israel
(bremler, farber.boris)@idc.ac.il

David Hay
Danny Hendler
Ben Gurion University, Israel
(davidhay, hendlerd)@cs.bgu.ac.il

Ternary content-addressable memories (TCAMs), which compare in parallel packet headers against all rules in a classification database, are increasingly used for high-speed packet classification.

TCAMs are not well-suited for representing rules that contain range fields. Such rules are represented by multiple TCAM entries and the range expansion they introduce can dramatically reduce TCAM utilization. This redundancy can be mitigated by making use of extra bits, available in each TCAM entry.

We present a novel scheme for constructing efficient representations of range rules, based on the simple observation that sets of disjoint ranges may be encoded more efficiently than sets of overlapping ranges.

Our technique splits the ranges between multiple layers each of which consists of mutually disjoint ranges. Each layer is then coded independently and assigned its own set of extra bits. An extensive comparative analysis on real-life classification databases establishes that our algorithms reduce the number of redundant TCAM entries caused by range rules by more than 60% as compared to best range-encoding prior art.

Why layering improves the performance?

Encoding and search key generation without layering

```
+-------+
| 001   |
| 10*   |
| 01**  |
+-------+
```

Observation: While encoding n arbitrary ranges may require n bits, only \( \log(n + 1) \) bits are required to encode n disjoint ranges.

Our Contributions

Novel scheme for constructing efficient representations of range rules, by making efficient use of TCAM extra bits.

- Algorithms for partitioning the ranges to sets of disjoint ranges (layers)
- Coding each layer independently with its own set of extra bits
- Supports hot updates for database changes
- Supports multiple range fields

Our empirical results show that all our algorithms reduce the average number of redundant TCAM entries required to represent a range rule decreases by more than 60% as compared to best range-encoding prior art.

Experimental Results and Comparative Analysis

- Settings: Real-life database, which is a collection of 120 separate rule files originating from various applications; approximately 223,000 rules that contain 280 unique ranges. 28% of the rules contain range fields and about half of these include the range [1024, 2161].
- Results: Either one of the four layering algorithms we analyze reduces the redundancy factor by 50%–70% (depending on the number of available extra bits) as compared with DRES, which is the best prior art range encoding algorithms. In the typical IPv4 TCAM-based classification databases, when 36 extra bits are available our scheme reduces the redundancy factor by either 62.2% or 67.8%, depending on whether the fall-back scheme used is prefix expansion or SROG coding, respectively.

Scheme Outline

```
Layering Stage
How to partition the ranges to sets of disjoint ranges?
```

```
Bit Allocation Stage
How many bits to give to each layer?
```

```
Encoding Stage
How to encode the rules and search keys?
```

```
Search Key: 110000100
```

Redundancy

The iterative algorithm Bit Auction. Each iteration is an auction, in which layers compete for the next available bit. If a layer \( L_j \) has already been assigned \( x_j \) bits, then assigning it additional \( k \) bits allows encoding \( L_j \)'s next \( 2^k-4 \) intervals. According to this we compute the per bit decrease in redundancy gained by allocating the next \( k \) bits to each layer and assign the next bit to a layer for which this quantity is maximal.

Encoding Stage

- Determining the code of each range: The ranges are encoded in decreasing weight order within each layer, according to the number of bits allocated to the layer. The extra bits corresponding to other layers are set to ‘*’. If a range fall out of the bit budget of the layer, all extra bits are set to ‘*’ and a fall-back scheme is used.
- Encoding rules with ranges: If the rule’s range is encoded – set ‘*’ in its original field and use the extra bits determined in previous step. Otherwise, use fall-back scheme and set ‘*’ in all extra bits.
- Encoding Search Keys: The value of the extra bits is determined by concatenating all the codes of the ranges the entry intersect and filling with 0-bits the bits corresponding to layers in which the entry does not intersect any range. The rest of the search key is unchanged.

Bit Allocation Stage

```
Search Key generation with a layered interval code
```

```
+-------+ +-------+
| 001   | | 001   |
| 01**  | | 10*   |
| 10**  | | 01**  |
+-------+ | +-------+
```

Search Key: 101011

Redundancy factor as a function of the number of extra bits for different layering schemes.