High Performance Construction of RecSplit Based Minimal Perfect Perfect Hash Functions

ESA 2023, Amsterdam

Dominik Bez, Florian Kurpicz, Hans-Peter Lehmann, Peter Sanders | September 4, 2023
Minimal Perfect Hashing

- Static set of $n$ keys
- Bijectively map keys to the first $n$ integers
- Recent idea: RecSplit [EGV20] (Esposito, Mueller Graf, Vigna)
- Push the boundaries of practical space usage
- Utilize modern processors and GPU
Bijections: RecSplit [EGV20]

- Brute-force
- Bit pattern indicates used hash values (single machine word)
- Store successful seed $s$
- Brute-force
- Bit pattern indicates used hash values (single machine word)
- Store successful seed $s$
RecSplit [EGV20]

- Hash keys to buckets
- Tree structure within buckets
  - Brute-force search for splitting hash function
  - Specific shape depending only on bucket size
- Small leaves of size \( \ell \leq 16 \)
  - Brute-force search for bijection hash function
Bijections: Rotation Fitting

- Split keys into two subsets
- Determine function values independently
- Cyclically “rotate” word $b$
- Store seed and rotation $s \cdot \ell + r$
- Test $\approx \ell$ times fewer seeds
**Bijections: Rotation Fitting**

- Split keys into two subsets
- Determine function values independently
- Cyclically “rotate” word $b$
- Store seed and rotation $s \cdot \ell + r$
- Test $\approx \ell$ times fewer seeds

![Diagram of bijections and rotation fitting](image)

1-bit hash function $h_s$ applied to $a$ and $b$:

$\begin{array}{cccc}
1 & 0 & 0 & 1 \\
0 & 1 & 0 & 0
\end{array}$

OR operation results in:

$\begin{array}{cccc}
1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1
\end{array}$
Bijections: Rotation Fitting

- Split keys into two subsets
- Determine function values independently
- Cyclically “rotate” word $b$
- Store seed and rotation $s \cdot \ell + r$
- Test $\approx \ell$ times fewer seeds
Bijections: Rotation Fitting

- Split keys into two subsets
- Determine function values independently
- Cyclically “rotate” word $b$
- Store seed and rotation $s \cdot \ell + r$
- Test $\approx \ell$ times fewer seeds

Diagram:

- 1-bit hash function
- Rotate $b$ by $r = 2$
- OR operation
Bijections: Rotation Fitting

- Split keys into two subsets
- Determine function values independently
- Cyclically “rotate” word $b$
- Store seed and rotation $s \cdot \ell + r$
- Test $\approx \ell$ times fewer seeds
CPU Parallelization

- Bit parallelism
  - Bit operations rotate all keys of a leaf
- SIMD parallelism
  - Each lane tries a different hash function seed
- Multi-Threaded parallelism
  - Calculate different buckets in parallel
Threads try different seeds

Groups of threads work on different tree nodes

2D grid of groups to calculate all trees with same shape

Streams to calculate different tree shapes in parallel
Construction with Rotation Fitting

![Graph showing speedup vs. bits per key for Brute force and Rotation fitting methods.]

- **Bits per key**
- **Speedup**

- **Brute force**
- **Rotation fitting**

September 4, 2023, Bez, Kurpicz, Lehmann, Sanders: High Perf. Constr. of RecSplit Based MPHFs

Institute of Theoretical Informatics
Multi-Threaded Construction

Throughput (MKeys/s) vs Bits/Key for 1 Thread and 16 Hyperthreads.

- BBHash [LRCP17]
- CHD [BBD09]
- PTHash [PT21a]
- PTHash-HEM [PT21b]
- RecSplit [EGV20]
- SicHash [LSW23b]
- SIMDRecSplit
## GPU Construction

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Method</th>
<th>Threads</th>
<th>Bits/key</th>
<th>Construction</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\ell = 16, b = 2000$</td>
<td>RecSplit [EGV20]</td>
<td>1</td>
<td>1.560</td>
<td>1175.4 $\mu s$/key</td>
<td>1 $\times$</td>
</tr>
<tr>
<td></td>
<td>SIMDRecSplit</td>
<td>16</td>
<td>1.560</td>
<td>27.9 $\mu s$/key</td>
<td>42 $\times$</td>
</tr>
<tr>
<td></td>
<td>GPURecSplit</td>
<td>GPU</td>
<td>1.560</td>
<td>1.0 $\mu s$/key</td>
<td>1175 $\times$</td>
</tr>
<tr>
<td>$\ell = 18, b = 50$</td>
<td>RecSplit [EGV20]</td>
<td>1</td>
<td>1.707</td>
<td>2942.9 $\mu s$/key</td>
<td>1 $\times$</td>
</tr>
<tr>
<td></td>
<td>SIMDRecSplit</td>
<td>16</td>
<td>1.708</td>
<td>12.3 $\mu s$/key</td>
<td>239 $\times$</td>
</tr>
<tr>
<td></td>
<td>GPURecSplit</td>
<td>GPU</td>
<td>1.709</td>
<td>0.5 $\mu s$/key</td>
<td>5438 $\times$</td>
</tr>
<tr>
<td>$\ell = 24, b = 2000$</td>
<td>GPURecSplit</td>
<td>GPU</td>
<td>1.496</td>
<td>467.9 $\mu s$/key</td>
<td>—</td>
</tr>
</tbody>
</table>
Conclusion

- New technique **Rotation Fitting**
- Heavy parallelization
  - Bits, Vectors, Cores, GPU
- Up to **5438 times faster** construction
- First to achieve **1.4x bits** per key
- 🍃/ByteHamster/GpuRecSplit

- Future work: Improve query performance
- New: ShockHash [LSW23a] for bijections

This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 882500).


References II

