# Worm Hash Map

## Time and Memory Benchmarking

with JMH

**Aleksandr Danilin**

May 20, 2020

# Outline

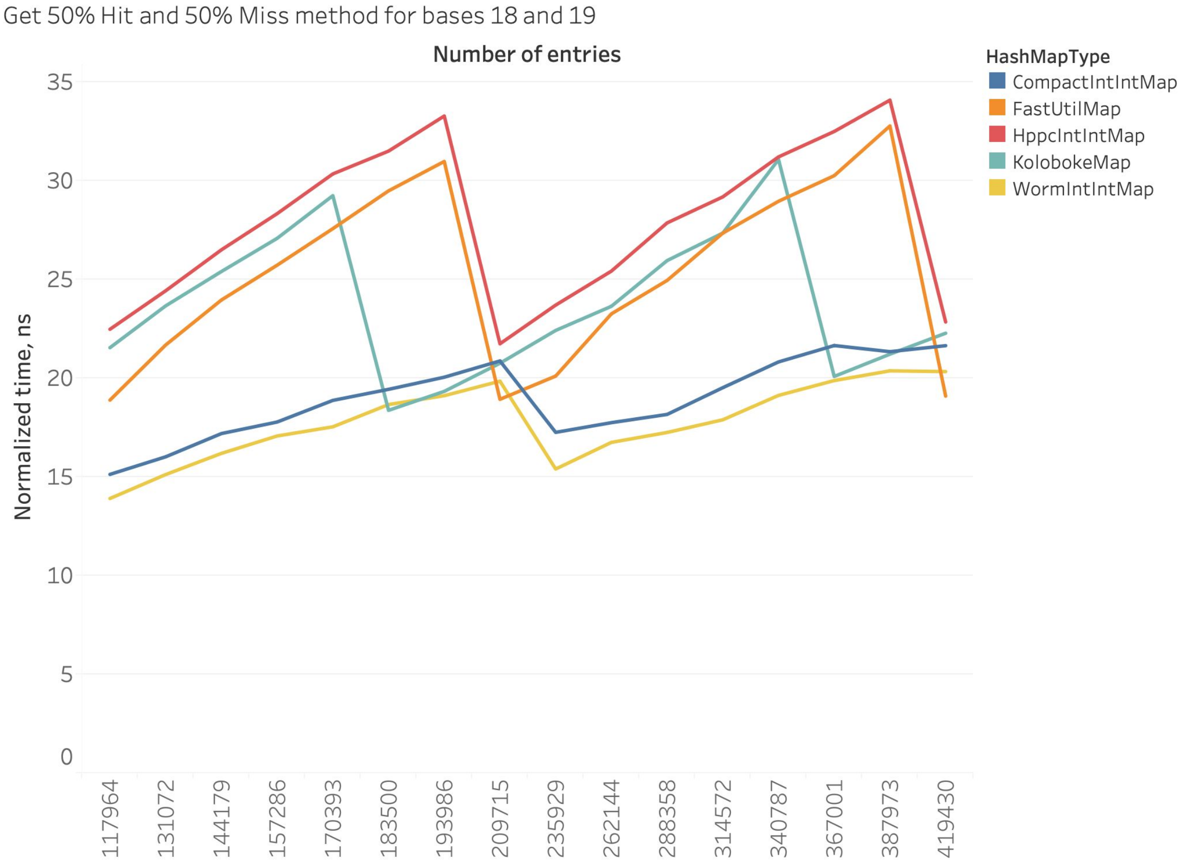
- Compact/Worm map optimizations
- Time optimization results
- Memory usage simulation
- Memory usage charts
- Overall results

# Compact map and its optimization branches

1. **Compact** - the original map;
2. **Worm** - compact map modified using a new hashing function, that is also used in FastUtil and Koloboke. Get method performs faster;
3. **WormReduced** - worm map with a change in Put method, which leads to faster Put and smaller load factors.

The main trade-off is speedup in exchange for smaller load factor.

# Get method optimization

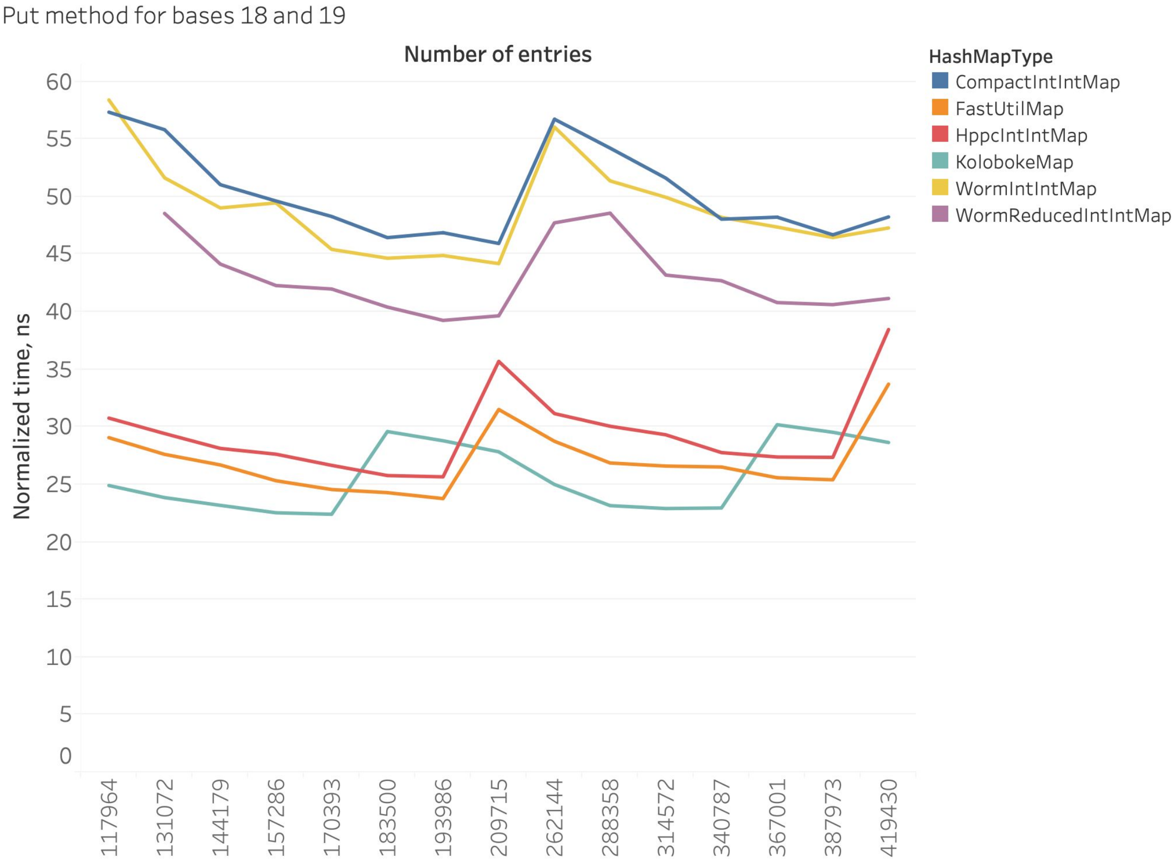


## Overall improvements:

Worm map is 6% faster than the Compact one.

Worm map is 35% faster than Koloboke, 43% faster than FastUtil, and 48% faster than HPPC.

# Put method optimization



**Time improvements:**

For size 157286 Compact and Worm maps is 95% slower than FastUtil map, 78% slower than HPPC.

For the same size WormReduced map is 67% slower compare to FastUtil, and 50% slower than HPPC.

# Memory Comparison - formula based approach

We didn’t find a reliable method that will measure actual memory usage of the maps.

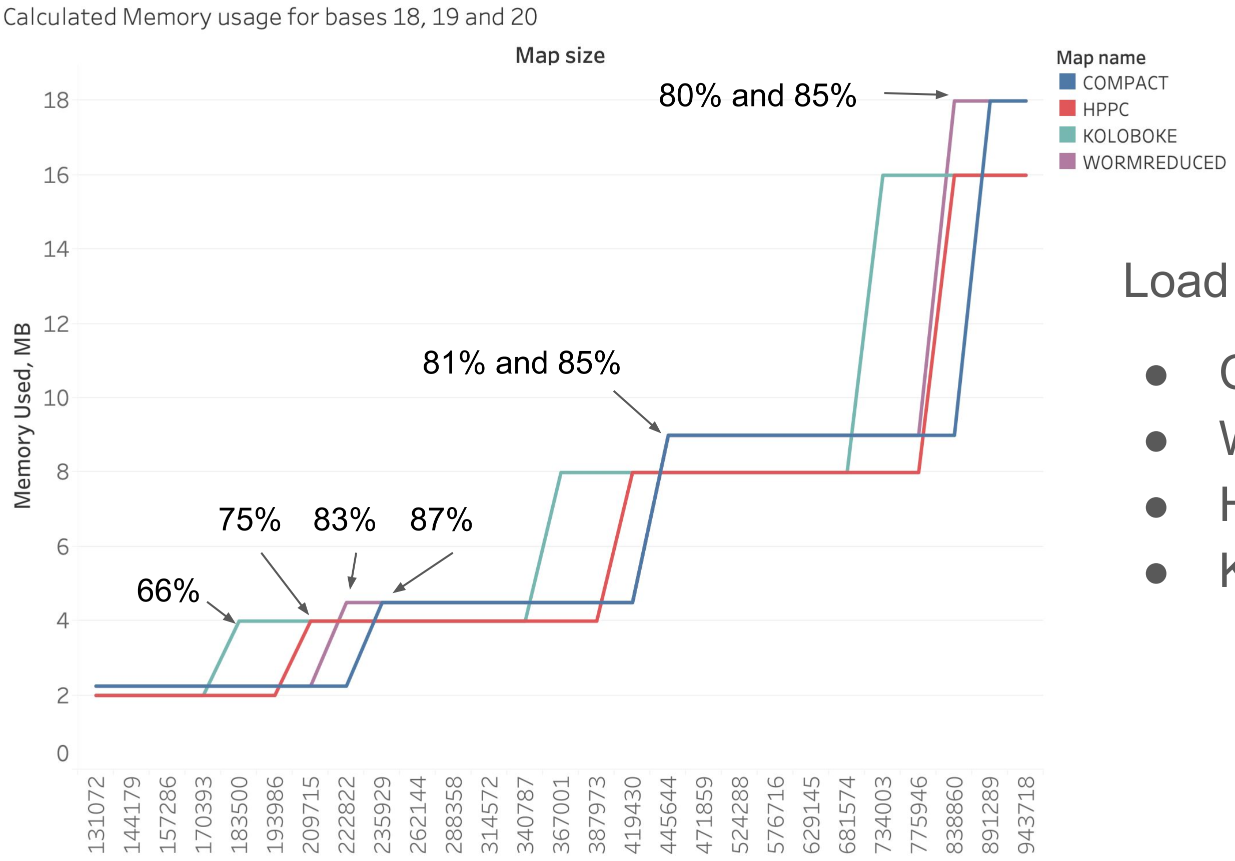
But we can simulate it using formulas derived from their code.

Where N - map capacity, K - key type size, V - value type size.

| N * (K + V) | N * (K + V + 1) |
|---|---|
| FastUtil | Compact and its branches |
| HPPC | Trove |
| Koloboke | |

In case of int-int entries JDK HashMap uses at least 4 times more memory.

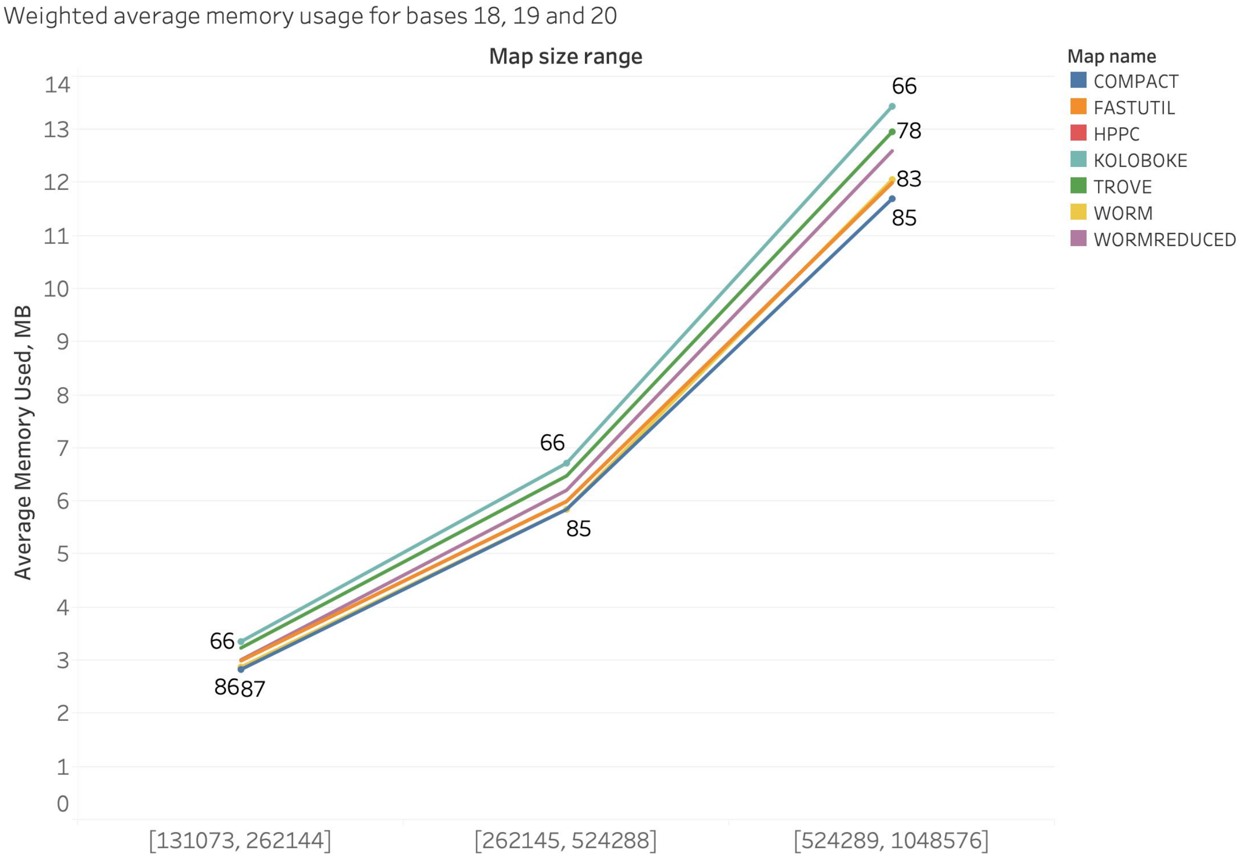
# Overall Memory usage



## Load Factors:

- Compact - [85% - 93%]
- WormReduced - [80% - 85%]
- HPPC, FastUtil - 75%
- Koloboke - 66%

# Average Memory Usage



**Chosen ranges:**

]2^(n-1), 2^n]

**Comparison results:**

Up to a million entries WormReduced is at most 6% bigger than FastUtil/HPPC.

At the same size range Compact and Worm maps are from 3% to 13% more efficient than FastUtil/HPPC.

# Overall results

| Measure\Map | Compact | Worm | WormReduced |
|---|---|---|---|
| Average Load Factor* | 87.9% | 87.2% | 83.1% |
| Get method | initial | 6% faster | 6% faster |
| Put method | initial | 4% faster | 15% faster |

| Measure\Map | HPPC | Worm | WormReduced |
|---|---|---|---|
| Average memory usage* | initial | 16% less | 11% less |
| Get method | initial | 48% faster | 48% faster |
| Put method | initial | 78% slower | 50% slower |

(*) for bases from 12 to 20