

# Right-sizing is hard, resizable BPF maps for optimum map size

*Thursday, August 27, 2020 9:00 AM (45 minutes)*

Right-sizing BPF maps is hard. By allocating for a worse case scenario we build large maps consuming large chunks of memory for a corner case that may never occur. Alternatively, we may try to allocate for the normal case choosing to ignore or fail in the corner cases. But, for programs running across many different workloads and system parameters its difficult to even decide what a normal case looks like. For a few maps we may consider using the BPF\_F\_NO\_PREALLOC flag, but here we are penalized at allocation time and still need to charge our memory limits to match our max memory usage.

For a concrete example, consider a sockhash map. This map allows users to insert sockets into a map to build load balancers, socket hashing, policy, etc. but, how do we know how many sockets will exist in a system. What do we do when we overrun the table?

In this talk we propose a notion of resizable maps. The kernel already supports resizable arrays and resizable hash tables giving us a solid grounding to extend the underlying data structures of similar maps in BPF. Additionally, we also have the advantage of allowing the BPF programmer to tell us when to grow these maps to avoid hard-coded heuristics.

We will provide two concrete examples where the above has proven useful. First, using the sockmap and sockhash tables noted above. This way we can issue a `bpf_grow_map()` indicating to the BPF map code more slots should be allocated if possible. We can decide using BPF program logic where to put this low-water mark. Finally, we will also illustrate how using resizable arrays can ensure the system doesn't run out of slots for the associated data in an example program. This has become a particularly difficult problem to solve with the current implementations where worse case can be severe, requiring 10x or more entries than the normal case. With the addition of resizable maps we expect many of the issues with right-sizing can be eliminated.

## I agree to abide by the anti-harassment policy

I agree

**Primary author:** FASTABEND, John (Isovalent)

**Presenter:** FASTABEND, John (Isovalent)

**Session Classification:** Networking and BPF Summit

**Track Classification:** Networking & BPF Summit