Prior work says GPU acceleration of packet processing is a good idea

**GPUs:**
- >10x raw compute power
- ~4x higher memory bandwidth

But many packet processing applications are not compute/memory intensive!

**Memory latency hiding is the key GPU advantage**

**GPUs hide memory latency by context switching**

**Can we hide latency for CPU programs?**
- CuckooSwitch [CoNEXT 13]: Group prefetching
- Grappa [U. Washington]: Context switching

**Assume programmer exposes parallelism:**

```c
for(i = 0; i < num_threads; i++) {
    /*
     * Do something for thread i,
     * independent of other threads.
     */
}
```

**Properties:**
- Processes batch of elements
- Elements independent of each other
- Can switch between elements!
- Annotations for expensive accesses

**G-Opt input code:**

```c
// Prefetch, save label, and Switch element
find(key *K, value *V) {
    int I = 0, mask = 0;
    for(i = 0; i < B; i++) {
        idx[I] = hash(K[i]);
        Ptr(ptr) = table[idx[I]].ptr;
        V[I] = *ptr;
    }
    labels[I] = &&end;
    mask = SET_BIT(mask, I);
    if(mask == (1 << B) - 1) return;
    I = (I + 1) % B;
    goto *labels[I];
}
```

**G-Opt makes CPUs more resource efficient than GPUs**

**Increases instructions executed, but IPC more**

**Throughput increase:**

**Code is online:** https://github.com/efficient/gopt