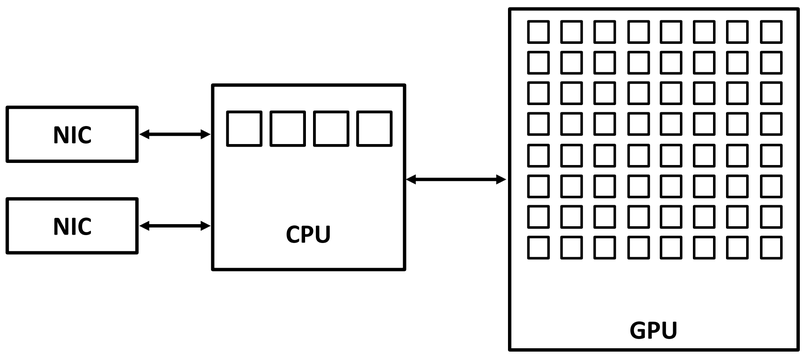
Raising the Bar for Using GPUs in Software Packet Processing

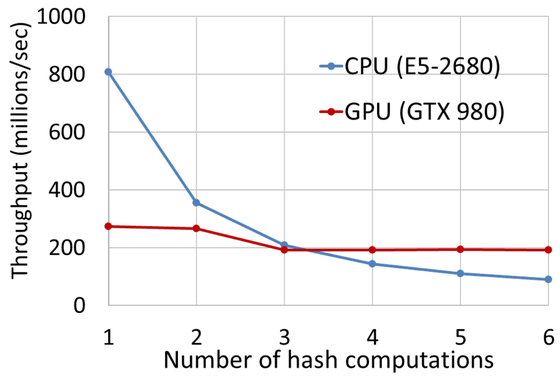
Anuj Kalia (CMU), Dong Zhou (CMU), Michael Kaminsky (Intel Labs), David Andersen (CMU)

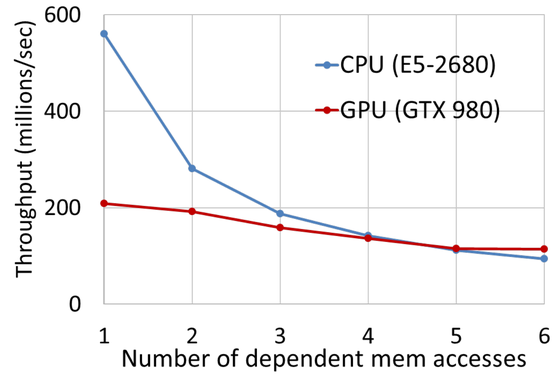
Prior work says GPU acceleration of packet processing is a good idea

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



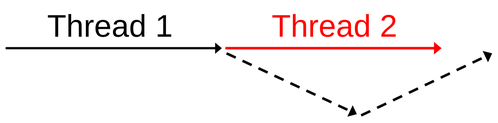
But many packet processing applications are not compute/memory intensive!





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:

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

G-Opt input code:

```
find(key *K, value *V) {
  int i;
  for(i = 0; i < B; i++) {
    int idx = hash(K[i]);
    _expensive(&table[idx]);
    value *ptr = table[idx].ptr;
    _expensive(ptr);
    V[i] = *ptr;
  }
}
```

G-Opt transform

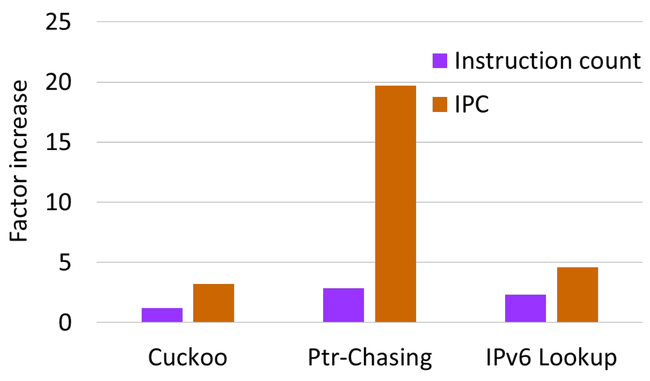
Local variables → Arrays
Annotations → Switching

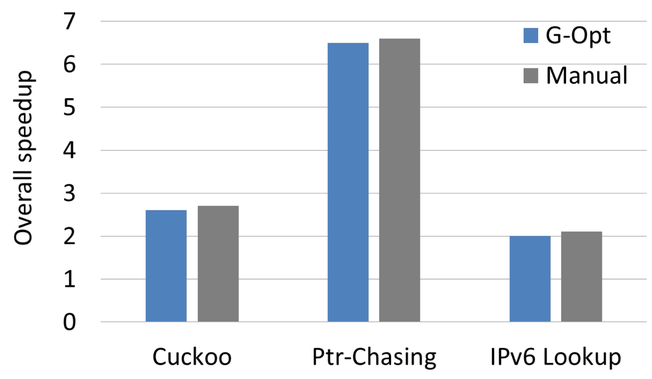
```
// Prefetch, Save label, and Switch element
#define PSS(addr, label) do {
  prefetch(addr); \
  labels[I] = &&label; \
  I = (I + 1) % B; \
  goto *labels[I]; \
} while(0);

find(key *K, value *V) {
  int idx[B];
  value *ptr[B];

  int I = 0, mask = 0;
  void *labels[B] = {label_0};

label_0:
  idx[I] = hash(K[I]);
  PSS(&table[idx[I], label_1);
label_1:
  ptr[I] = table[idx[I]].ptr;
  PSS(ptr[I], label_2);
label_2:
  V[I] = *ptr[I];

end:
  labels[I] = &&end;
  mask = SET_BIT(mask, I);
  if(mask == (1 << B) - 1) return;
  I = (I + 1) % B;
  goto *labels[I];
}
```

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

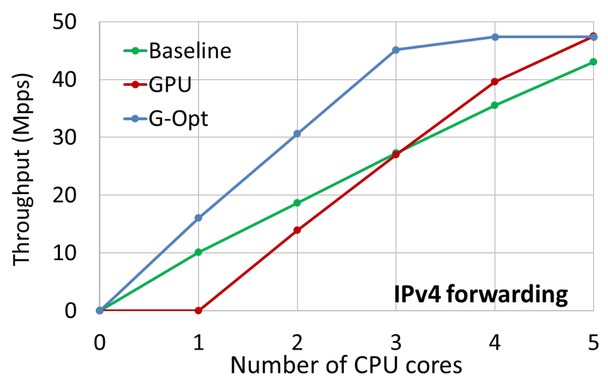
G-Opt makes CPUs more resource efficient than GPUs

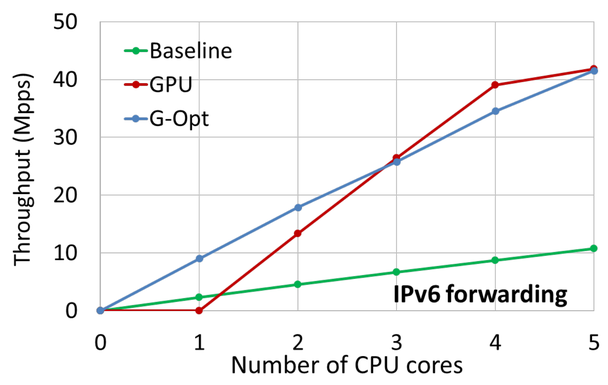
Increases instructions executed, but IPC more

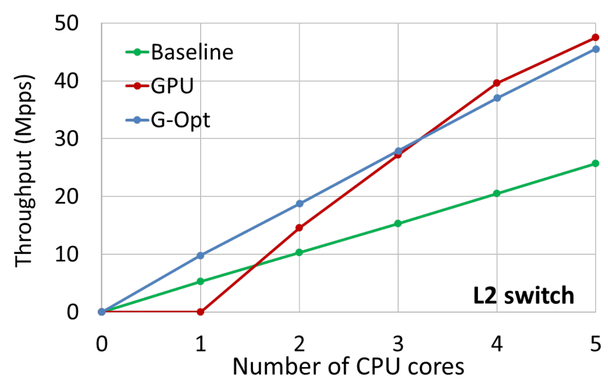


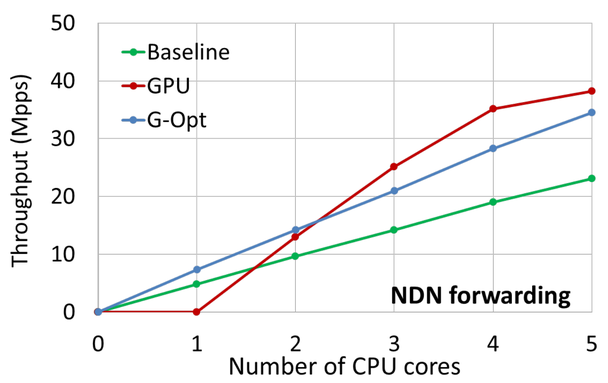


Throughput increase:









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

Intel Science & Technology Center for Cloud Computing
Carnegie Mellon University
Georgia Tech
intel
PRINCETON UNIVERSITY
UC Berkeley
UNIVERSITY of WASHINGTON