1. **Shorten Page Table to reduce accesses**

2. **Keep PTEs in the cache for quicker walks**

Every Walk’s a Hit: Making Page Walks Single-Access Cache Hits

Chang Hyun Park, Ilias Vougioukas, Andreas Sandberg, and David Black-Schaffer

---

**1 Background**

- TLB reach not scaling with DRAM
- TLB misses resolved with Page table walks
- Page walk caches enable skipping over levels
  - 1.1-2.5 (avg. 1.3) accesses per walk for workloads of up to 8 GB size

**2 Prior work: require large contiguous memory**

- Address Translation with Prefetching (ASAP)
- Elastic Cuckoo Page Tables (ECPT)

**3 Flatten the Page Table to reduce accesses**

- Use large pages (2 MB) as page table nodes
- Leverage existing OS support for large pages
- 2 MB page tables can flatten 2-levels of PT
- Flatten L4+L3 and L2+L1 → 2 memory accesses
- Page walker cache helps skip L4+L3 → 1 memory access
- Fallback to 4 KB PT when 2 MB not available

**4 Prioritize keeping PTE in the cache**

- Prioritize PT → cache PTE instead

**5 Results**

- Gem5 Simulator
- System-call Emulation mode
- Modeled after Intel Skylake uArch

---

*1 Use Schall et al., High Contiguous, Technical Report 2017. Coverage for Intel Skylake uArch
2 Margera et al., MICRO ’19
3 Margera et al., MICRO ’20
4 chang.hyun.park@it.uu.se
5 DOI: 10.1145/4189532.4189533
6 chang.hyun.park@it.uu.se
7 Paper 10.1145/4189532.4189533