

Introduction

We explore mechanisms that dynamically move data within caches by controlling page placement in main memory. The key innovation is the use of a shadow address space to allow hardware control of data placement in the L2 cache while being largely transparent to the user application and off-chip world. This allows the hardware and OS to dynamically manage cache capacity per thread, and optimize placement of data shared by multiple threads.

Motivation

- In future multi-cores, large amounts of delay and power will be spent accessing data in large L2/L3 caches.
- Multi-core processors will share last-level caches, hence managing these caches to avoid conflicts is important.
- Workloads will be mixed - multiple single threaded programs will execute simultaneously with multi-threaded programs.
- Shared cache management at fine-granularity (cache line) incurs large overhead, hence manage at page-granularity.

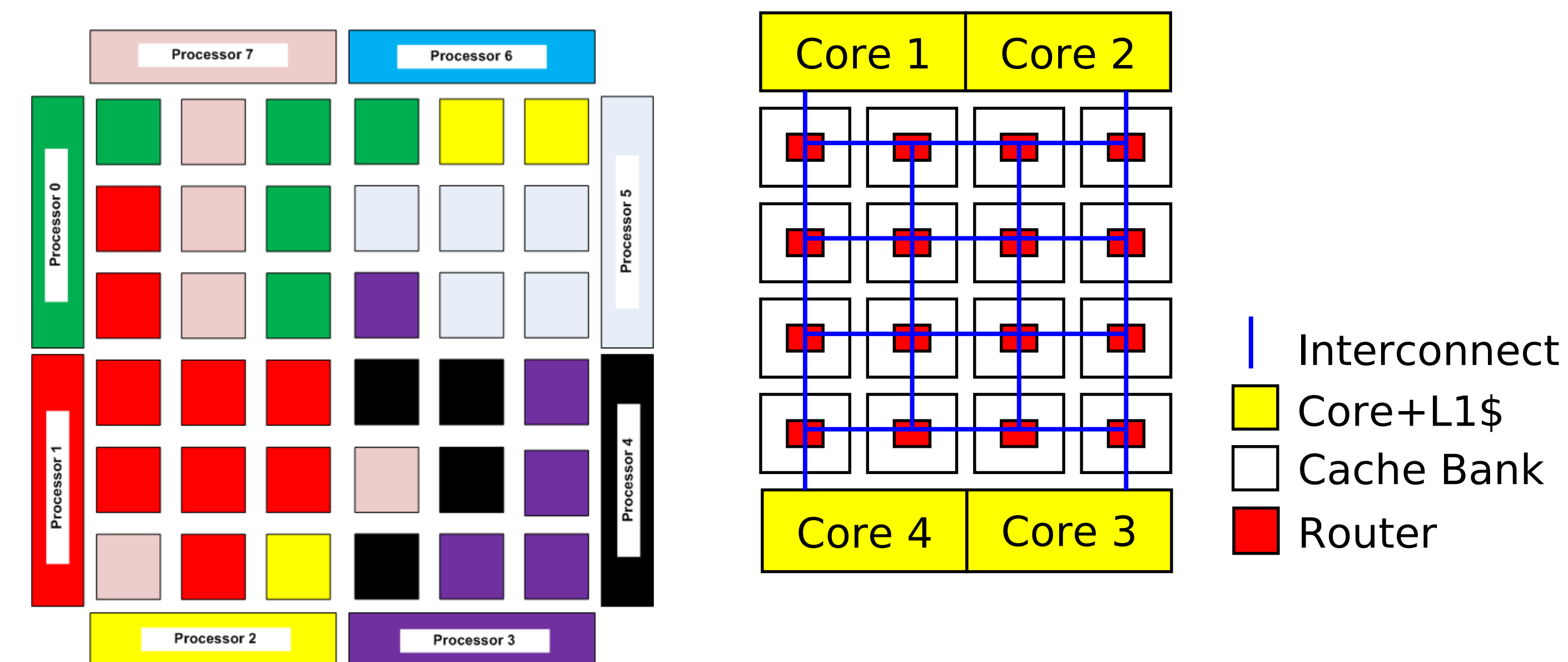
Goals

- Manage cache capacity and data placement to reduce conflicts and access time.
- Use dynamic profiling to make allocation and placement decisions.

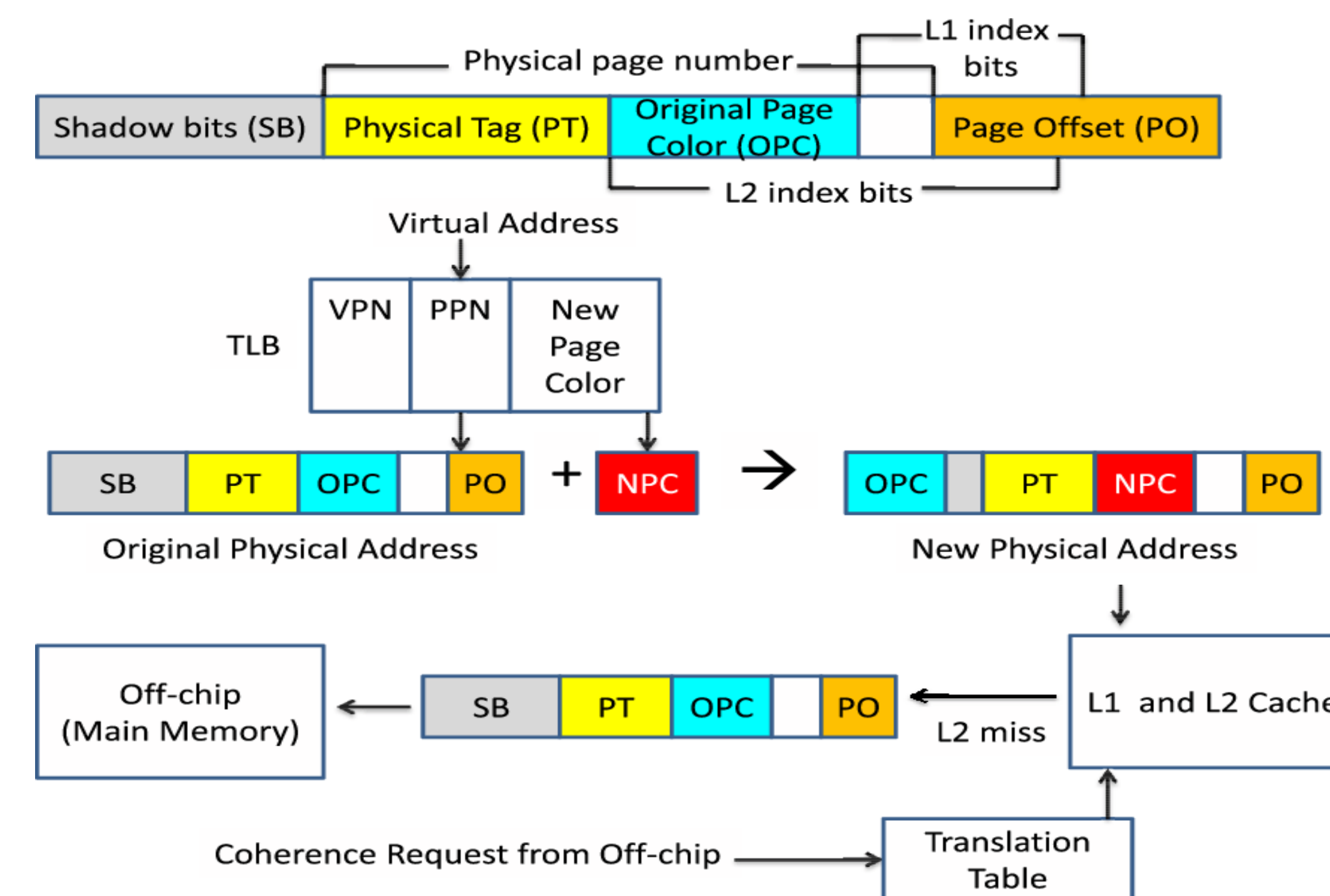
Method

- Introduce new level of indirection between main memory and cache addresses for flexible cache management.

Baseline Processor Model

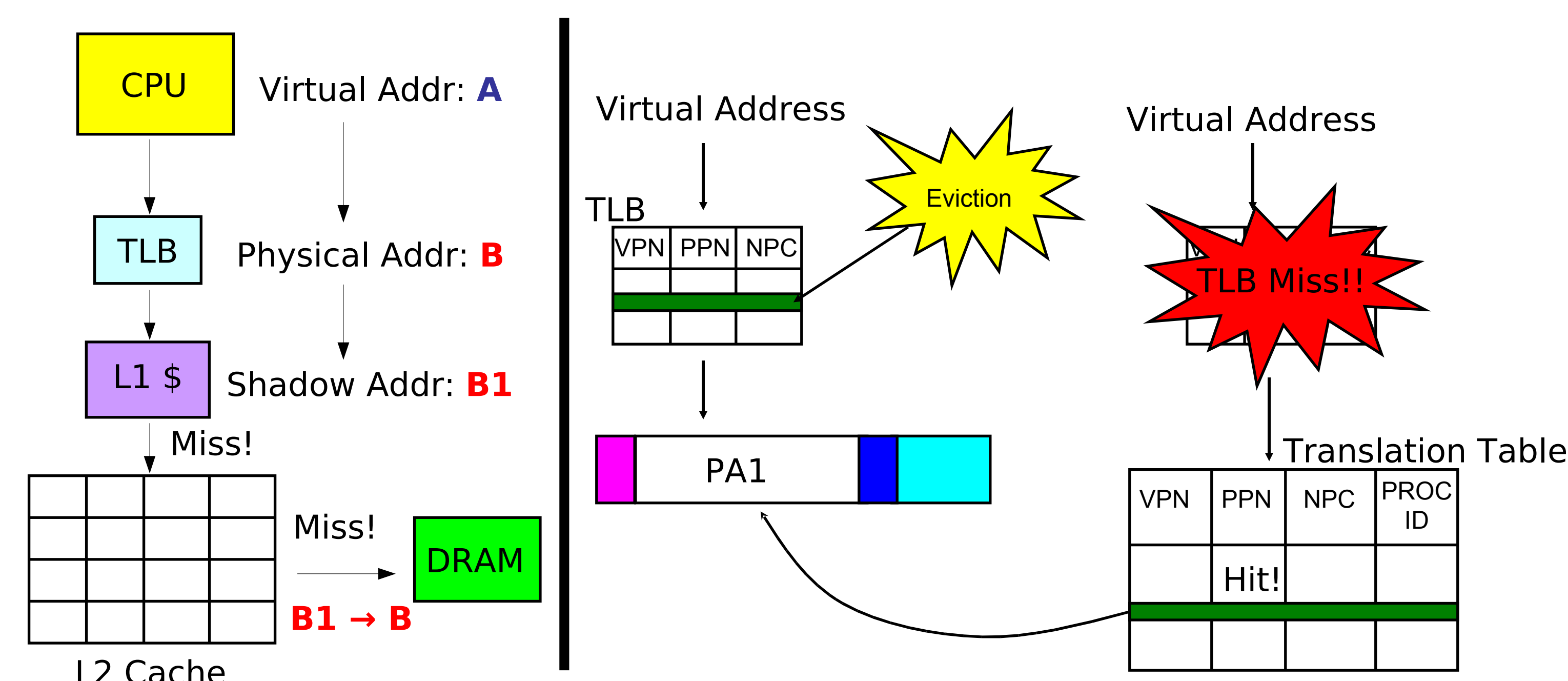


Proposed Mechanism



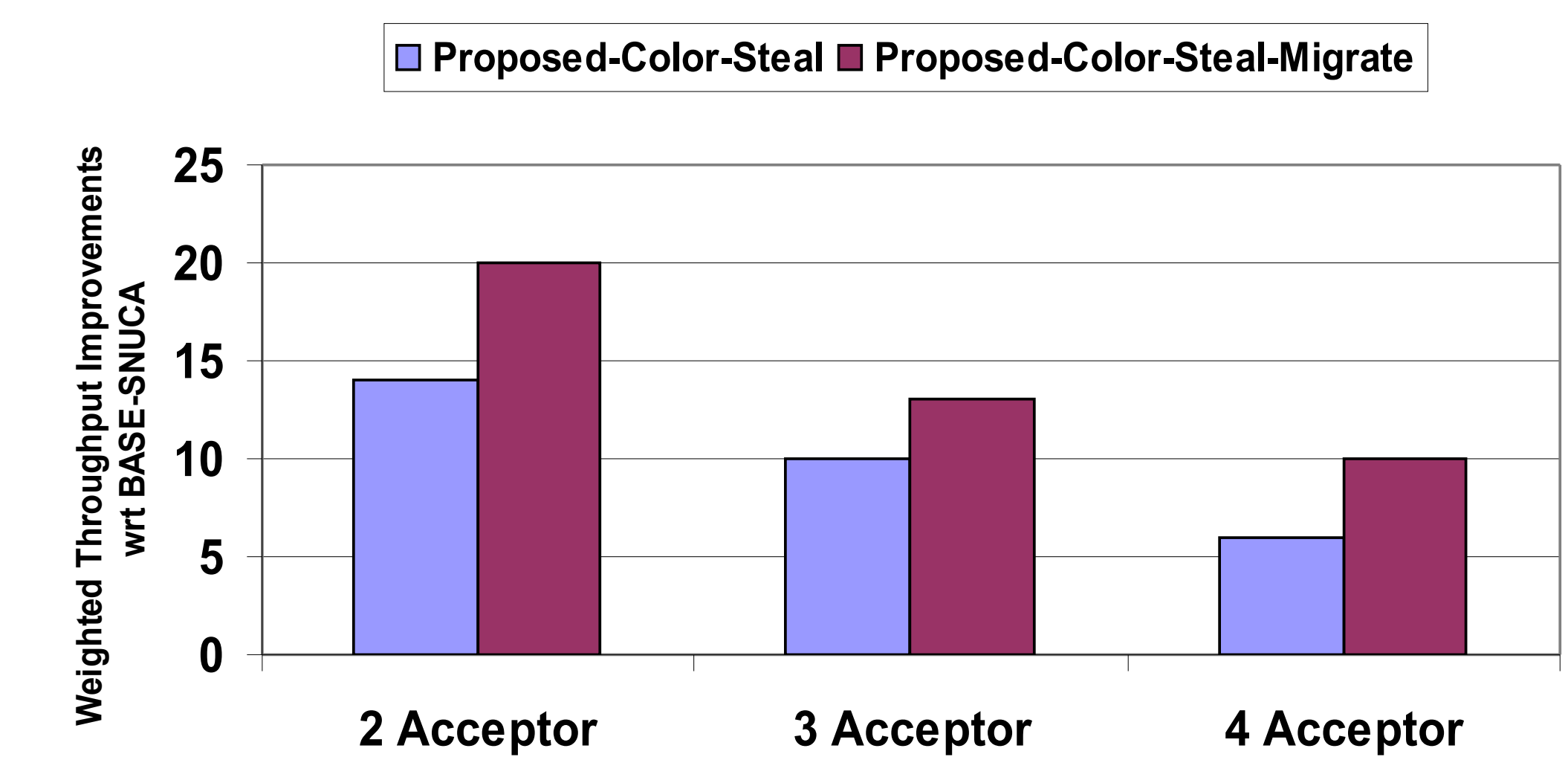
- Introduce a new level of indirection between main memory addresses by managing on-chip addresses at page granularity. TLB, and a **translation-table** keep track of new on-chip addresses.

Modified Cache Look-up Mechanism

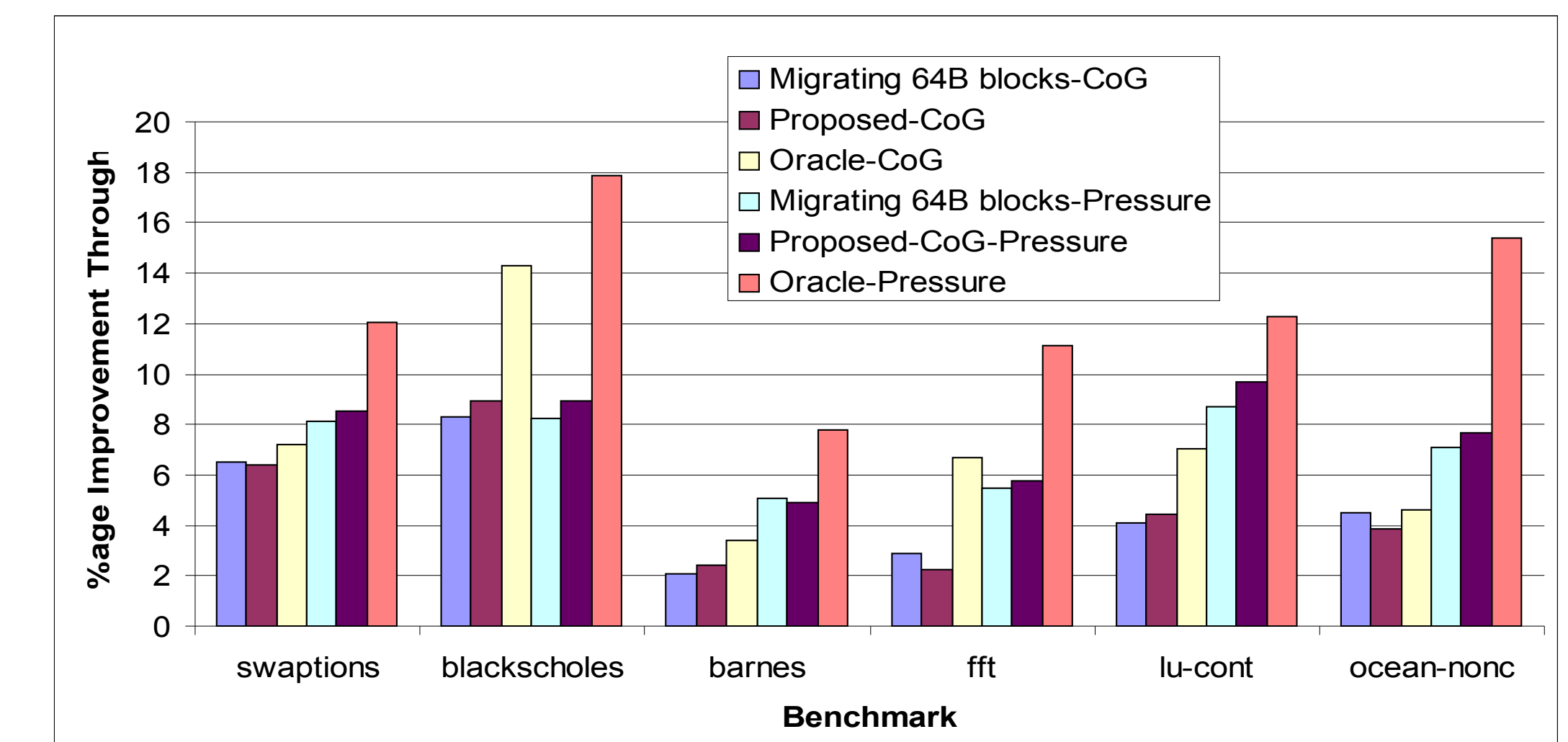


Results

➤ Multi-Programmed Workloads



➤ Multi-Threaded Workloads



- On average, 10 to 20% improvement for multi-programmed workloads and 8% improvement for multi-threaded workloads.

Conclusion

- A combined hardware-software approach with low overheads.
- Last Level cache management at page granularity.
- Use of page colors and shadow addresses for capacity and locality management

Pros and Cons

- Reducing wire delays.
- Localize private data.
- Optimal placement of shared data.
- Allows for fine-grained partition of caches.
- Translation Table is an overhead.