

Exclusive Control for Compound Operations On Hardware Transactional Memory

Keisuke MASHITA[†], Anju HIROTA[†], and Tomoaki TSUMURA[†]
[†]Nagoya Institute of Technology, Japan

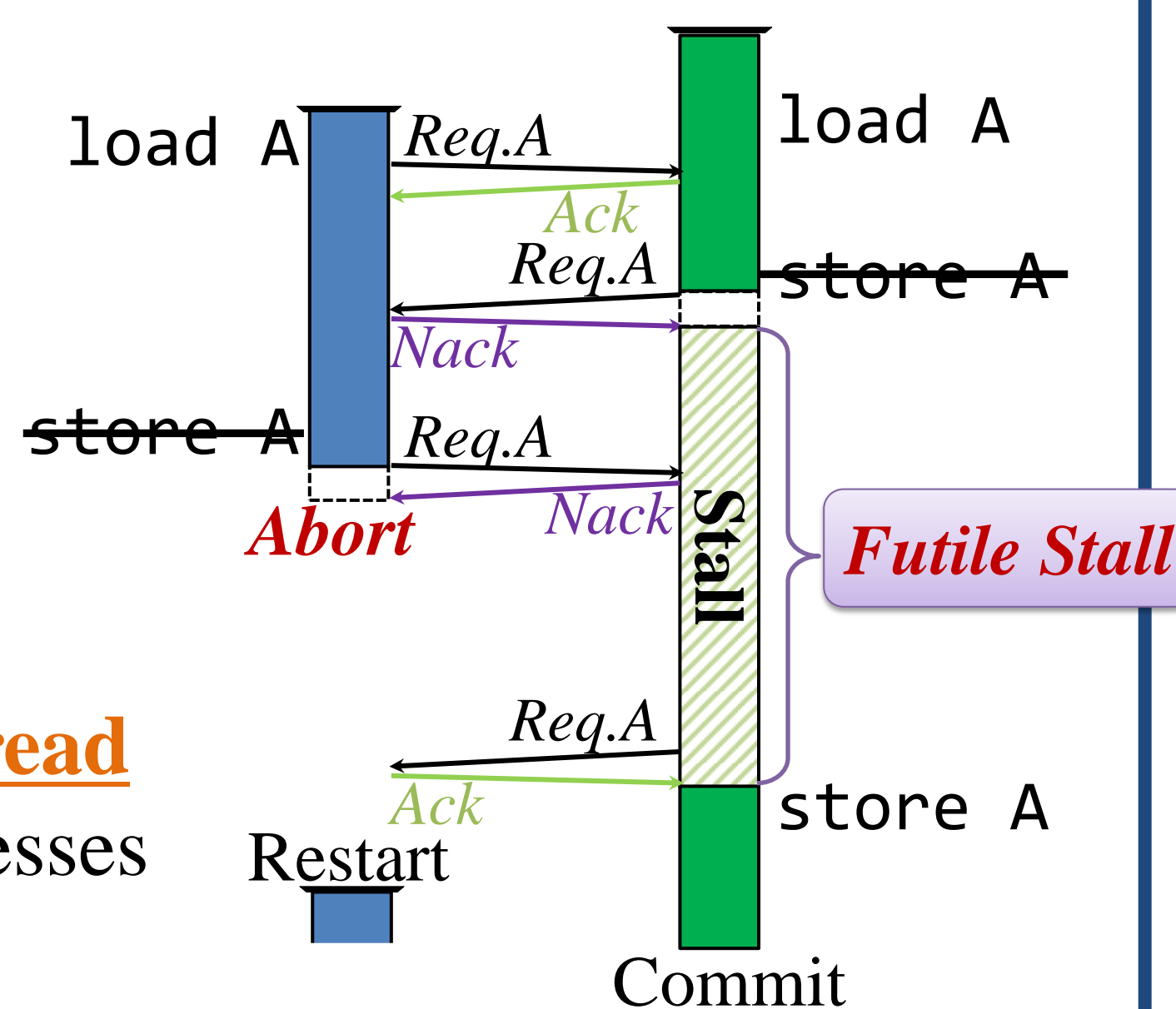
Summary

- **Hardware Transactional Memory (HTM)** is a promising mechanism for multi/many-core programming.
- A read variable will often be written before long, and this severely degrades the performance of TMs.
- We propose a transaction scheduling for mitigating this with very simple implementation.
 - The total execution cycles are reduced **72.2%** at a maximum, and **17.5%** on average.
 - The required additional hardware cost is only **512 Bytes**.

Compound Ops lead to Futile Stalls

Typical HTM scheduling

- When a thread tries to access a shared variable, the thread sends a request to detect a conflict.
- If a thread detects deadlock, the thread aborts its transaction.
- **Read-after-read (RaR) accesses** cause no conflict. However, many **read** accesses are followed by **write** accesses to the same addresses.



As a result, *Futile Stall* is caused even if an RaR access is allowed.

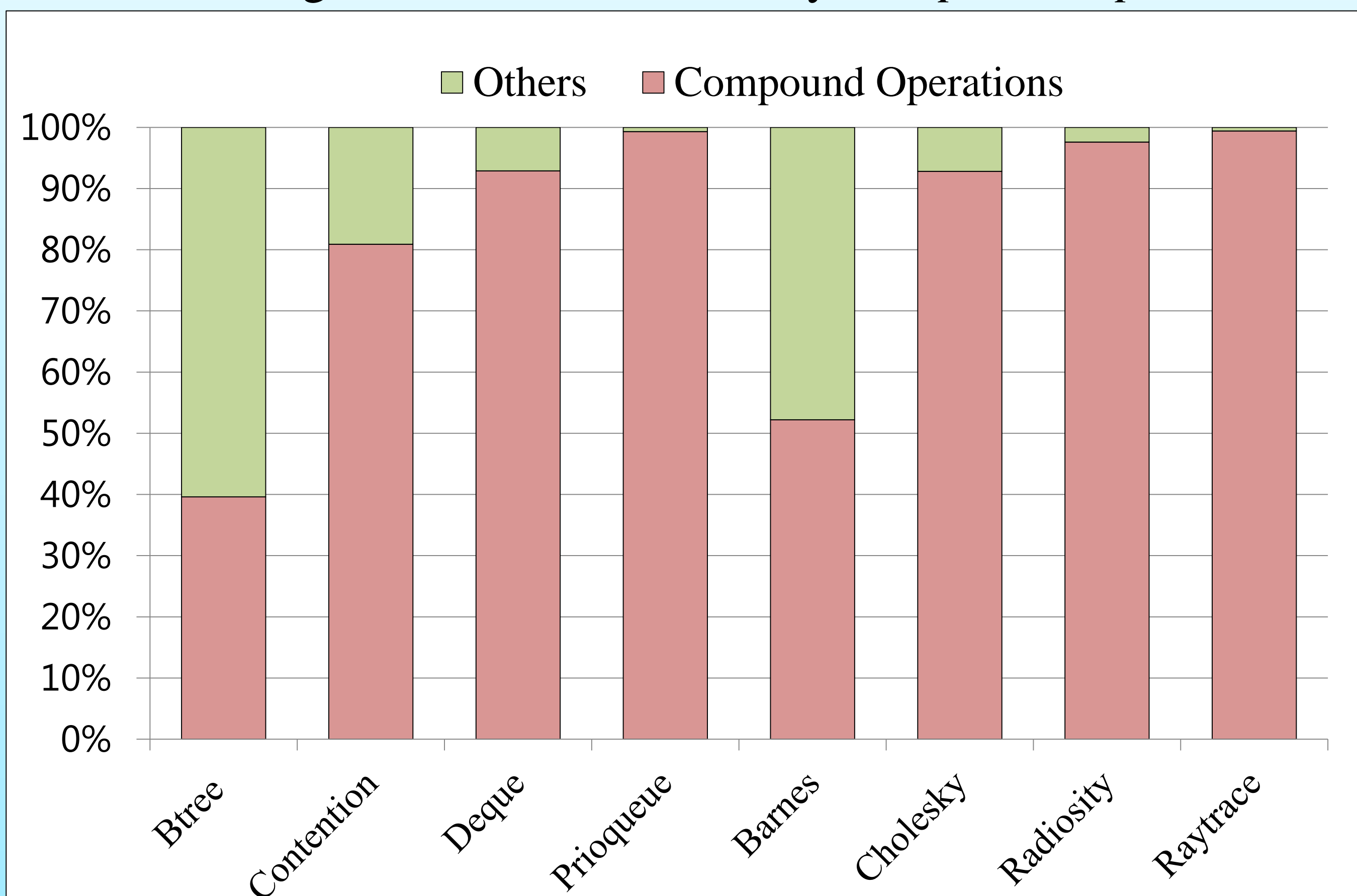
- Many Futile Stalls are caused by such an access pattern. (e.g. in **Compound Operations**)
 - Many transactions have such *compound operations*. (e.g. increment, decrement, and compound assignment operation)

The part of the transaction in Prioqueue

```
...
BEGIN_TRANSACTION
if (array[key] != index){ /* array[key] is read */
    array[key] = index; /* array[key] is written */
}
COMMIT_TRANSACTION
...
```

- Many aborts are caused by *Compound Operations*.
 - *Compound Operations* degrade the performance of HTM.

Percentage of the aborts caused by *Compound Operations*



We aim to solve these problems by a transaction scheduling with practical light-weight implementation.

Managing w/ a Flag in Each Cache Line

- We propose a very simple implementation.
 - Only an additional 1-bit flag (called **C-bit**) is required for each cache line.

L1Cache			
C	R	W	addr
0	0	0	-

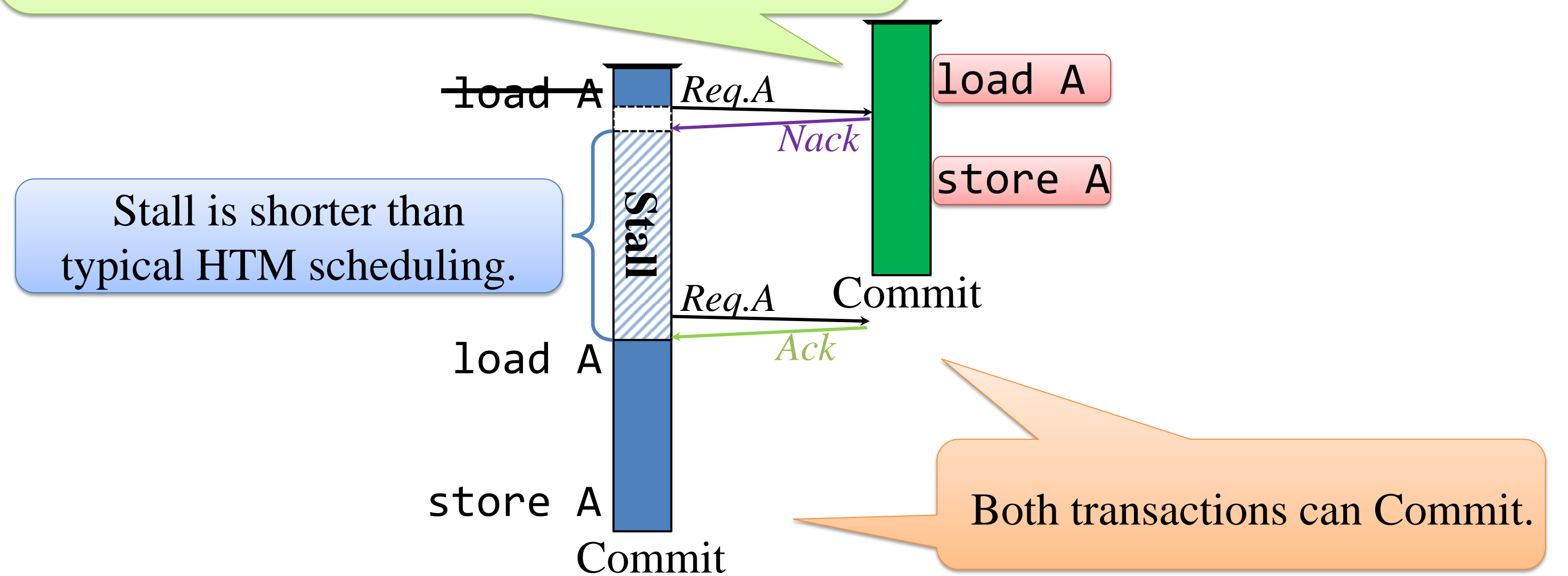
C-bit represents that the address has been accessed by a compound operation

Each cache line has these bits for conflict detection on general HTM

- If an RaR access to an address is expected to be followed by a write access, **the transactions are serialized**.

test the C-bit and detect a compound operation

C	R	W	addr
1	1	0	A



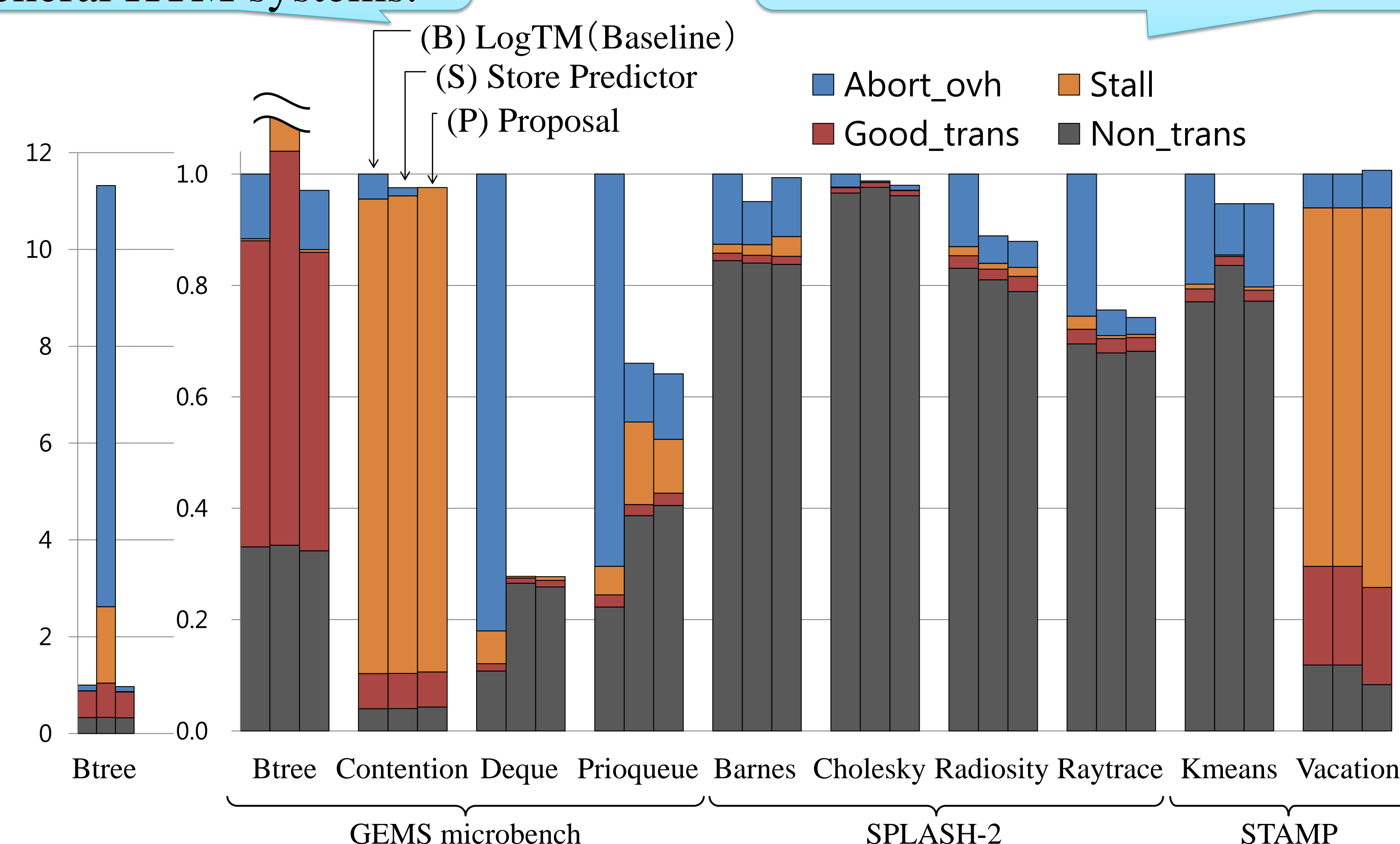
Performance Evaluation

Simulator

- Simics : Full system simulator (SPARC-V9, Solaris10)
- GEMS : Memory system simulator

LogTM is one of the most general HTM systems.

The execution cycles of the slowest thread in 16 threads.



The proposed scheduling prevents *Futile Stalls* and aborts. Especially, in Deque, abort is never caused with our scheduling.

Comparison between our proposal and *Store Predictor**

- *Store Predictor* is one of the existing work for solving bad influence of RaR accesses.
 - Unlike our proposal, even once an address is managed by *Store Predictor*, all accesses to the address are stalled.
 - This leads to serious performance degradation in Btree, because of unnecessary serialization.

* J. Bobba et al : Performance Pathologies in Hardware Transactional Memory, Proc.34th ISCA