BITS Pilani
Pilani Campus

Transactional Memory

K Hari Babu
Department of Computer Science & Information Systems

Transactions

- New paradigm to simplify programming
  - instead of lock-unlock, use transaction begin-end
- Can yield better performance; Eliminates deadlocks
- Programmer can freely encapsulate code sections within transactions and not worry about the impact on performance and correctness
- Programmer specifies the code sections they’d like to see execute atomically – the hardware takes care of the rest (provides illusion of atomicity)

Transactional memory

- Transactional memory is an alternative to lock-based synchronization
- Transactional memory is like database transactions where all shared memory accesses and their effects are either committed all together or discarded as a group
- All threads can enter the critical region simultaneously
    - If there are conflicts in accessing the shared memory data, threads try accessing the shared memory data again or are stopped without updating the shared memory data
    - Therefore, transactional memory is also called a lock-free synchronization
- Transactional memory can be a competitive alternative to lock-based synchronization.

Transactional memory

- A transactional memory system must hold the following properties across the entire execution of a concurrent program:
- Atomicity
  - All speculative memory updates of a transaction are either committed or discarded as a unit.
- Consistency
  - The memory operations of a transaction take place in order. Transactions are committed one transaction at a time.
- Isolation
  - Memory updates are not visible outside of a transaction until the transaction commits data.

Transactions

- A transaction executes speculatively in the hope that there will be no conflicts
- Can replace a lock-unlock pair with a transaction begin-end the lock is blocking, the transaction is not programmers can conservatively introduce transactions without worsening performance

```
lock (lock1)                 transaction begin
   read  A                      read  A
   operations                   operations
   write A                      write A
unlock (lock1)               transaction end
```

Hardware vs. software

- Hardware transactional memory (HTM) systems may comprise modifications in processors, cache and bus protocol to support transactions
- Software transactional memory (STM) provides transactional memory semantics in a software runtime library or the programming language and requires minimal hardware support (typically an atomic compare and swap operation, or equivalent)

References

- https://www.ibm.com/docs/en/xffbg/121.141?topic=fortran-transactional-memory
- TransactionalMemory - GCC Wiki (gnu.org)

Q&A

innovate achieve lead

BITS Pilani
Pilani Campus

Thank You