Virtual World consistency: a new consistency condition for STMs

Damien IMBS
damien.imbs@irisa.fr

PhD advisor: Michel RAYNAL

Workshop *What Theory for Transactional Memories?*

ASAP team, IRISA, Rennes, FRANCE

Summary

- Comparison between databases and STMs
- A first condition specific to STMs: Opacity
- Similarities between STMs and message-passing systems
- Virtual World consistency
- Conclusion

IRISA

Comparison between databases and STMs

Comparison between databases and STMs (1)

- Transactions are sets of reads, writes and local computations that can commit or abort (they can abort in order to allow concurrency)
- In databases, requirements are placed only on committed transactions
- Relational algebra is not expressive enough to allow aborted transactions to be problematic
- Usually, serializability or strict serializability (linearizability) are required from committed transactions

IRISA

Comparison between databases and STMs (2)

- STMs are general purpose: divisions by 0, infinite loops are possible
- Example without constraints on aborted transactions:
  - Initially, $A = 2$ and $B = 1$.
  - $A = B$ defines an inconsistent state of the shared memory.

Transaction 1:	Transaction 2:
(1) $A \leftarrow A * 2; B \leftarrow B * 2$	(2a) $z \leftarrow A$
	(2b) $c \leftarrow 1/(z - B)$

  - If (2a), (1), (2b), Transaction 2 will execute a division by 0 before aborting

A first condition specific to STMs: Opacity

A first criterion: Opacity (1)

Goal

- These problems can be solved by sandboxing, but at a high cost
- The goal of opacity is to forbid a transaction to read a set of values that is not consistent, avoiding the need to sandbox transactions
- It has first been introduced informally by Dice, Dolev and Shavit and has then been named and formalized by Guerraoui and Kapałka

Dice D., Shalev O. and Shavit N., Transactional Locking II, DISC’06.
Guerraoui R. and Kapałka M., On the Correctness of Transactional Memory, PPoPP’08.

A first criterion: Opacity (2)

- Intuitively, opacity requires that, when excluding the write operations of aborted transactions, the set of all transactions be linearizable
- Opacity prevents transactions from reading an inconsistent set of values, but also forces transactions that will abort to read mutually consistent sets of values
- It is sufficient but not necessary:

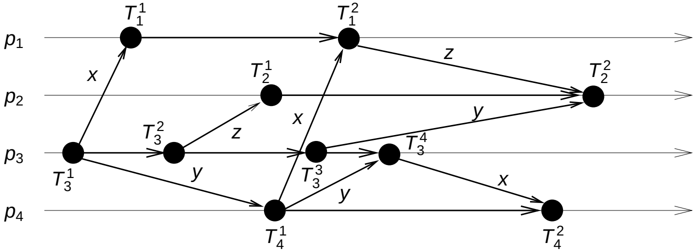
  The same goal can be attained with a weaker criterion

IRISA

Similarities between STMs and message-passing systems

Similarities with message-passing systems (1)

- Transactions can be seen as groups of “receive” and “send” events, where values are seen as messages



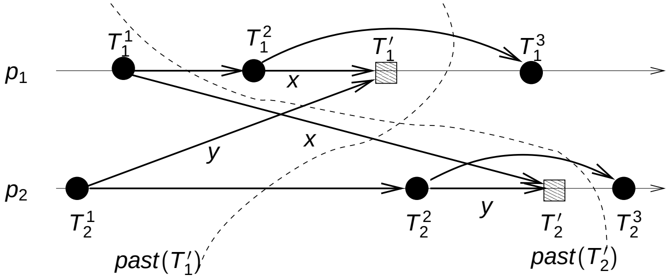
Similarities with message-passing systems (2)

- In a given STM execution, we define the partial order $\rightarrow_{PO}$ as follows. $T_1 \rightarrow_{PO} T_2$ if:
  - **Process order:** both T_1 and T_2 are issued by the same process, and T_1 is a committed transaction issued before T_2
  - **Read-from relation:** there is an object X whose value written by T_1 has been read by T_2 – equivalent to the message exchange relation
  - **Transitivity:** $\exists T : (T_1 \rightarrow_{PO} T) \wedge (T \rightarrow_{PO} T_2)$
  - Note: an aborted transaction doesn't have any successor in the $\rightarrow_{PO}$ relation
- $\rightarrow_{PO}$ is the STM equivalent of the "happen before" relation (causal order) in message-passing systems

IRISA

Causal past of a transaction

- The *causal past* of a transaction T, denoted $past(T)$, is the set including T and all the T' such that $T' \rightarrow_{PO} T$.



- Note: if T is an aborted transaction, it is the only aborted transaction contained in $past(T)$

Virtual World consistency

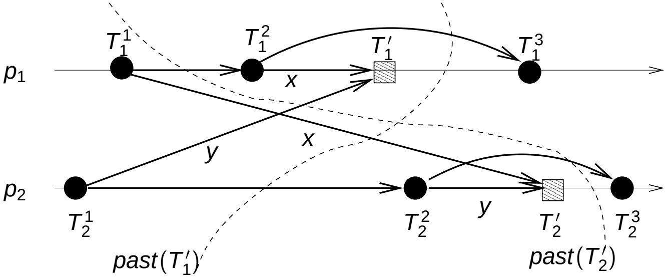
Virtual World consistency

- The goal is to obtain a consistency criterion weaker than opacity, but that also prevents transactions from reading an inconsistent set of values
- The idea is that different aborted transactions don't have to to observe the same witness sequential execution
- Virtual world consistency still requires that the set of committed transactions be linearizable
- For each aborted transaction T, the requirement is that $past(T)$ accepts a legal linearization that respects $\rightarrow_{PO}$

Imbs D. and Raynal M., On the consistency conditions of software transactional memories. *Tech Report #1917*, IRISA, 2008.

IRISA

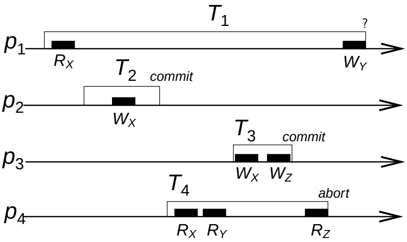
Example



- T_1' sees T_1^2 before T_2^2 – it reads y's value from T_2^1
- T_2' sees T_2^2 before T_1^2 – it reads x's value from T_1^1
- But considered individually, each transaction observes a consistent state of the shared memory

Virtual World vs Opacity

- Virtual world consistency allows committing more transactions than opacity



- With opacity, $T1$ must be aborted ($T4$ sees $T2$ before $T1$).
- With virtual world consistency, $T1$ can be committed.

Conclusion

Conclusion

- Virtual world consistency:
  a valid and useful consistency criterion
- A new view on STM systems