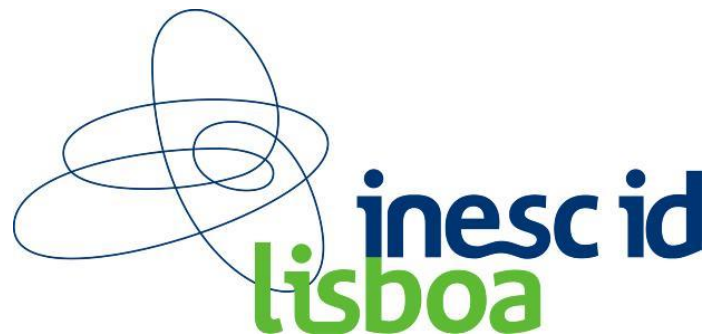


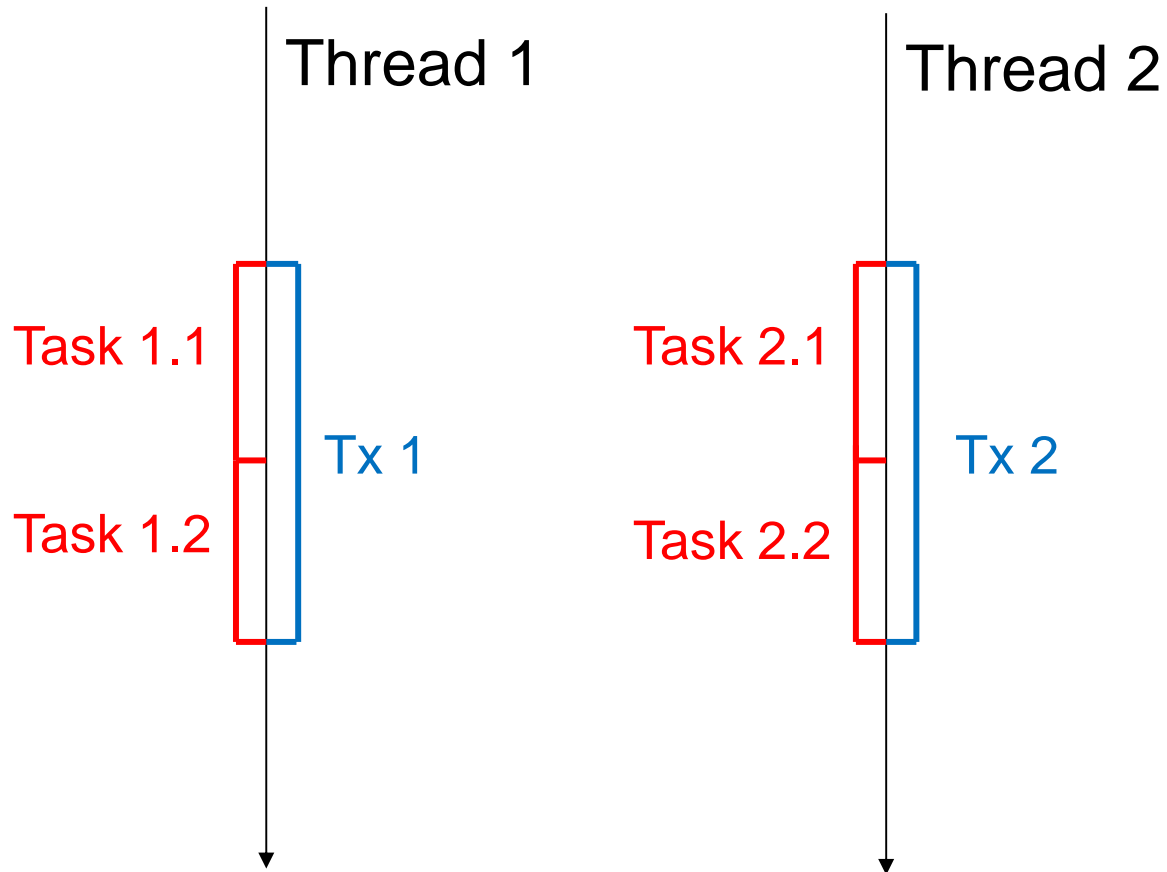
# (Towards) TLS+STM offloading to Helper Threads

Ricardo Filipe, João Barreto

<http://www.gsd.inesc-id.pt/project-pages/specSTM>



# Previous Work: Unified TLS+STM runtime

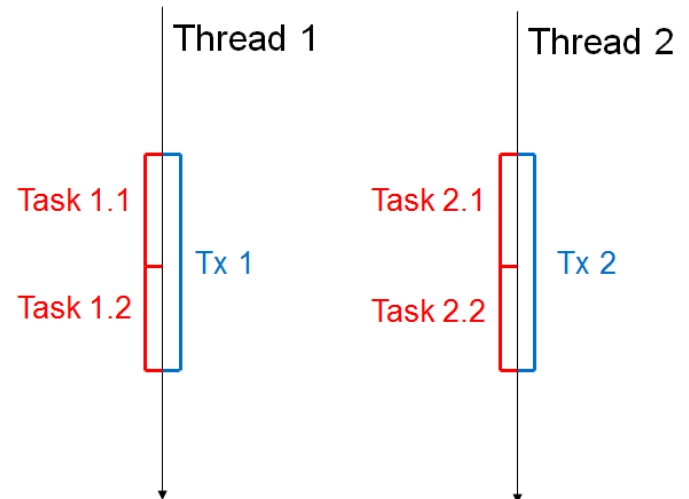


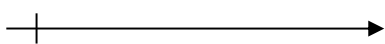
Core 1    Task 1.1 begins & Tx 1 begins  
+————→

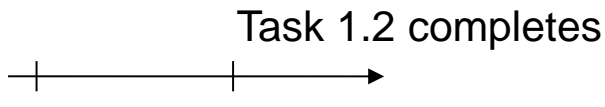
Core 2    Task 1.2 begins  
+————→

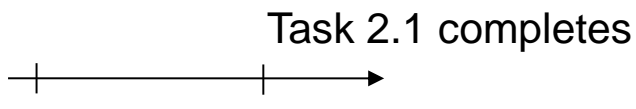
Core 3    Task 2.1 begins & Tx 2 begins  
+————→

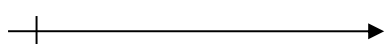
Core 4    Task 2.2 begins  
+————→

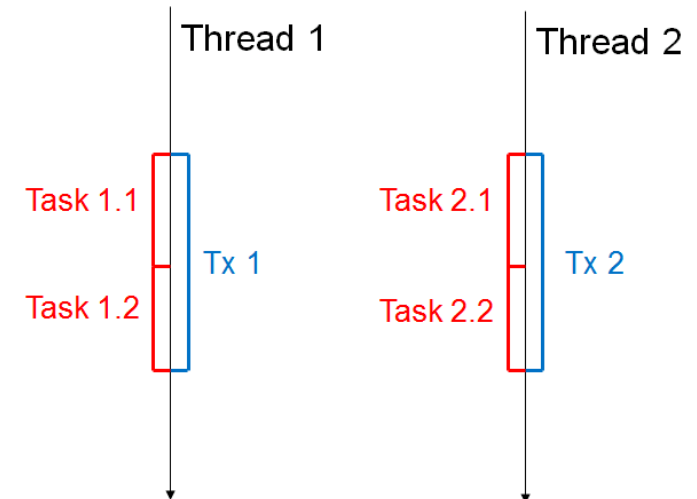


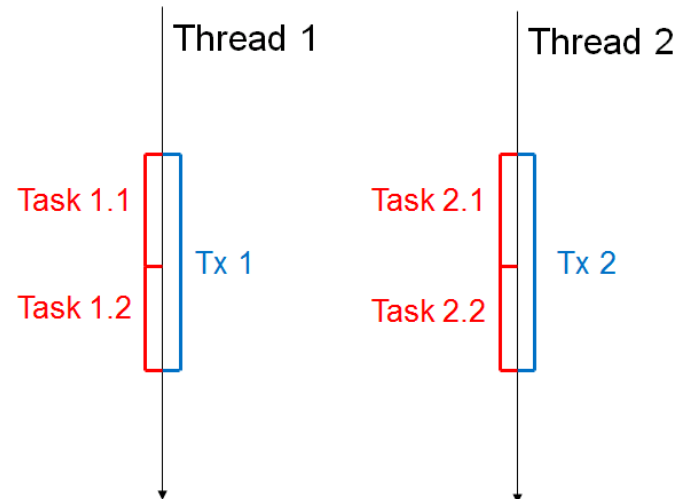
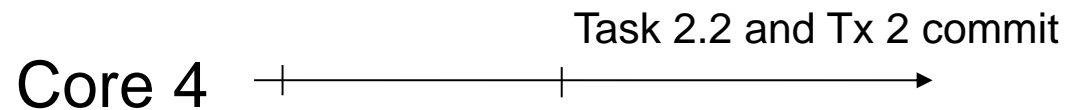
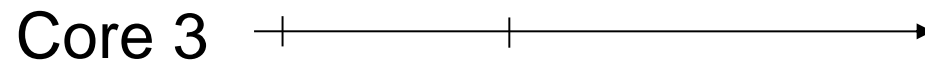
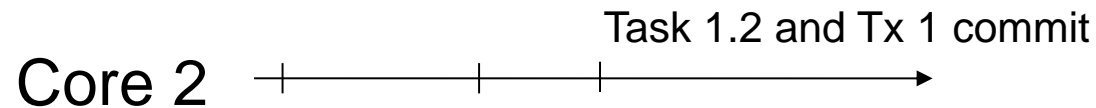
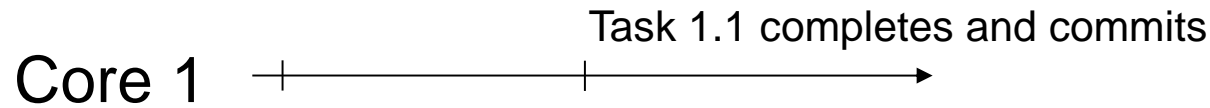
Core 1 

Core 2 

Core 3 

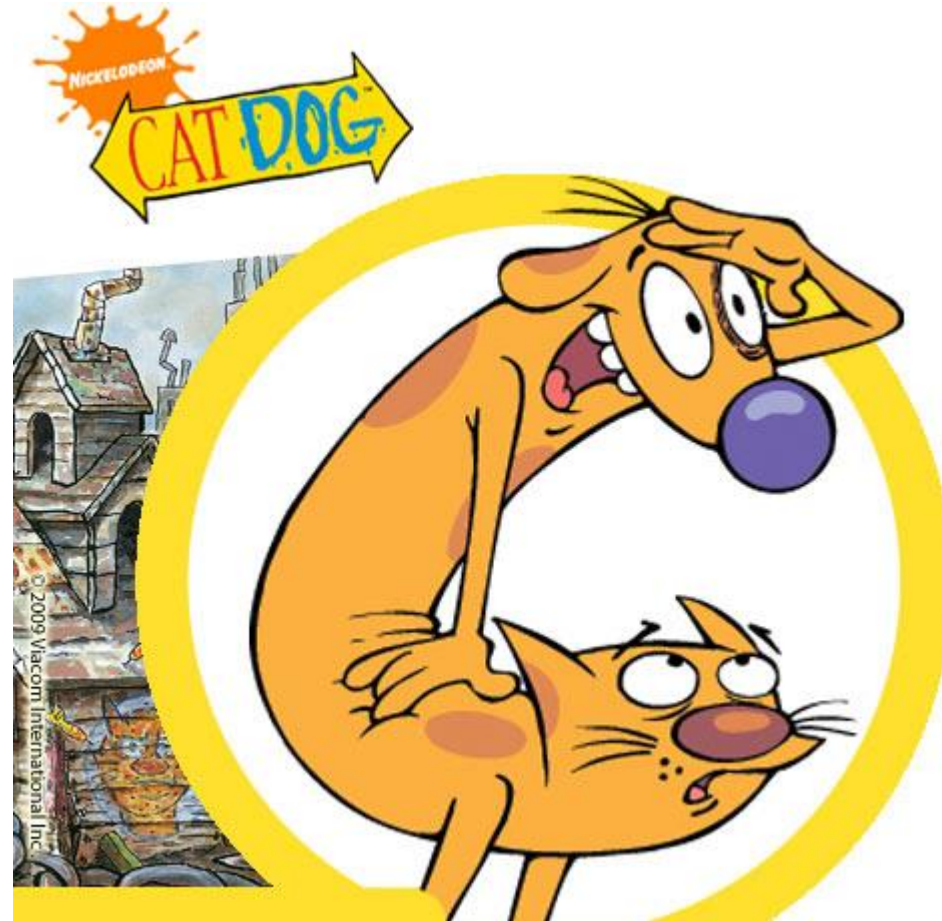
Core 4 



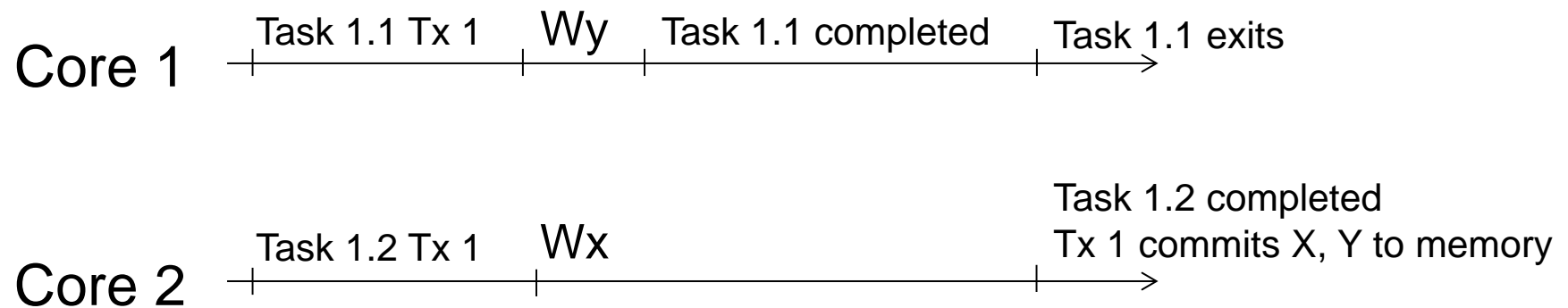


# TLSTM: The first unified TLS+STM runtime

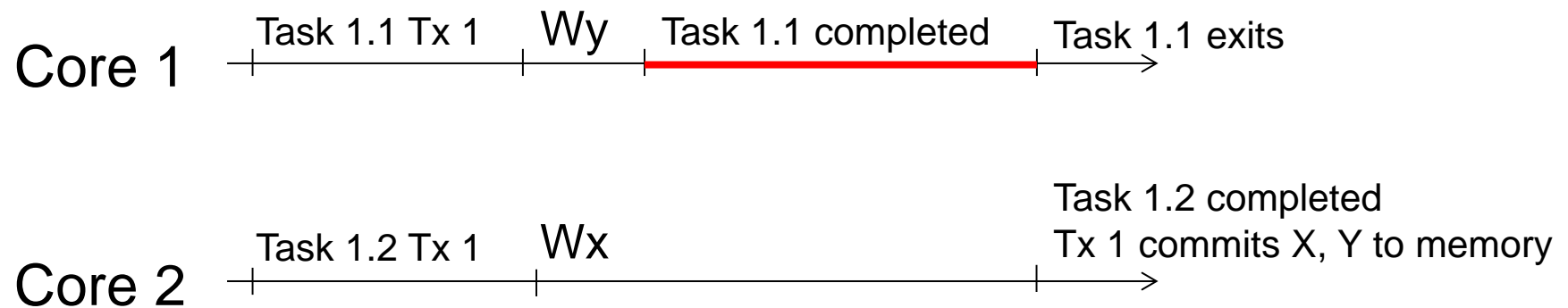
technology  
from seed



- Leveraging STM with TLS features
  - Transaction commit takes place in the last task of the transaction, after all of its tasks have completed



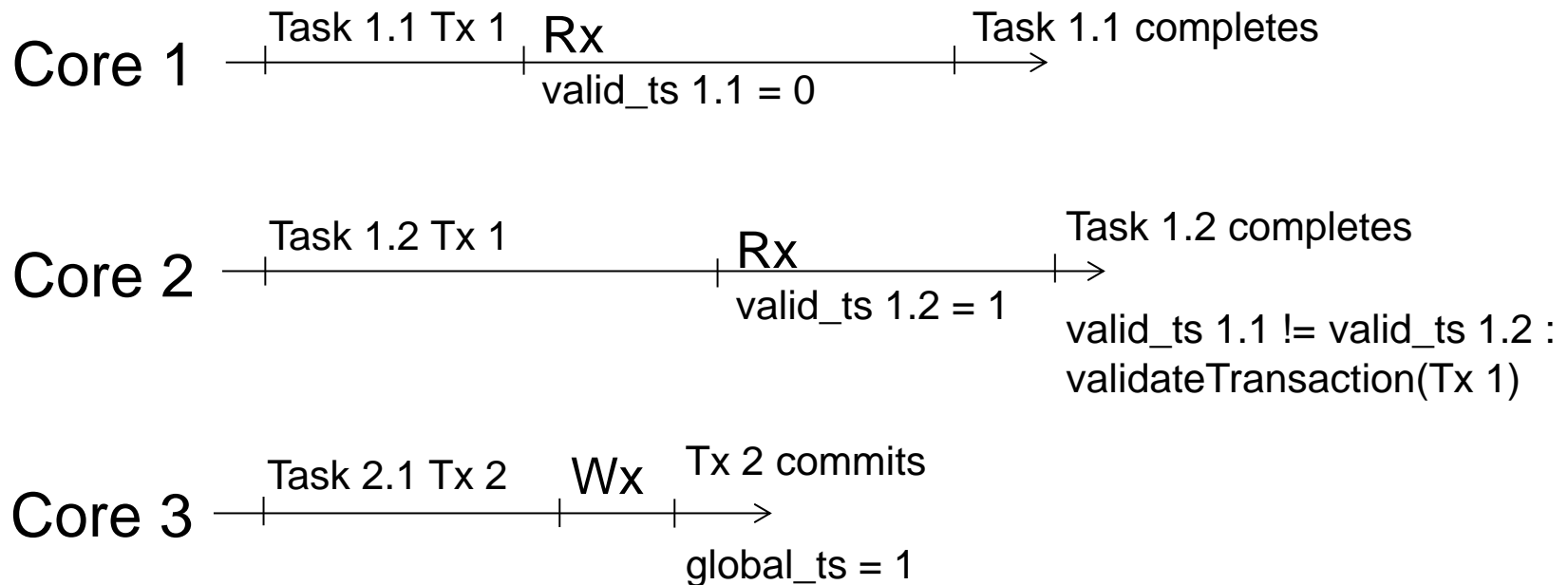
- Leveraging STM with TLS features
  - Transaction commit takes place in the last task of the transaction, after all of its tasks have completed
  - Core 1 could be working on another task/transaction, but it is busy waiting until it can exit the current task





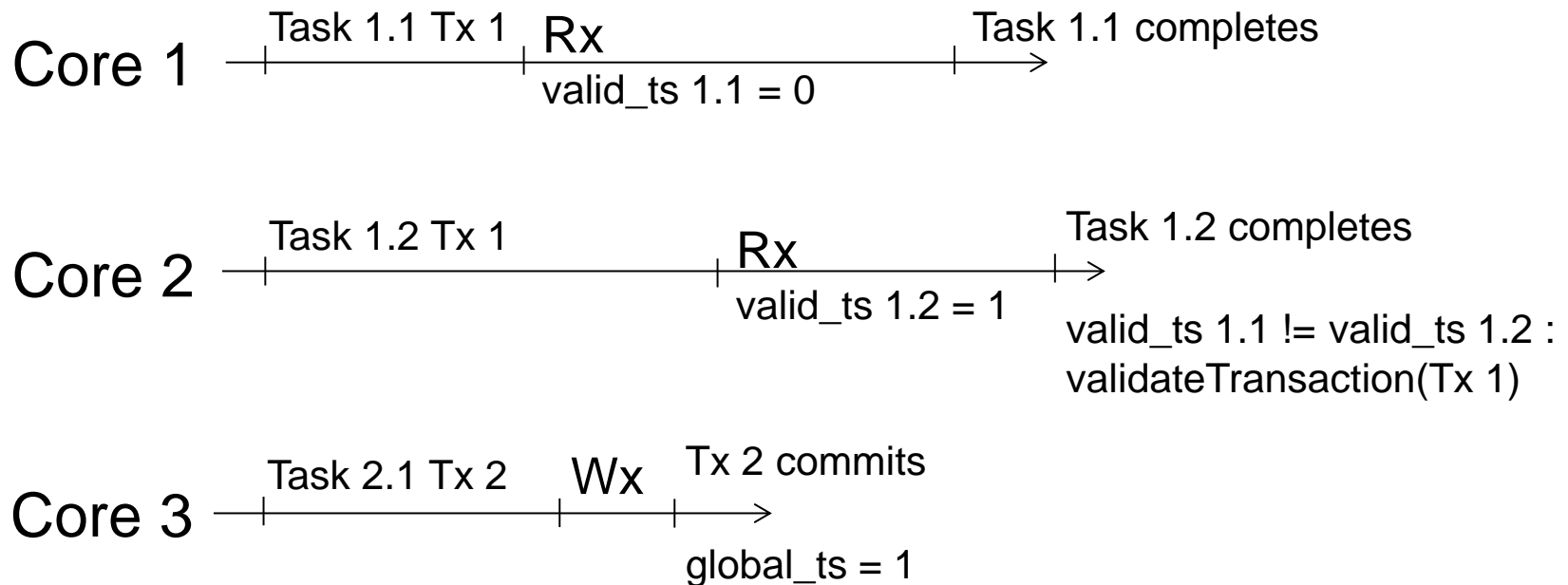
# Problems of TLSTM: Atomic operations on several tasks

- Leveraging STM with TLS features
  - Transaction validation is done separately by each task

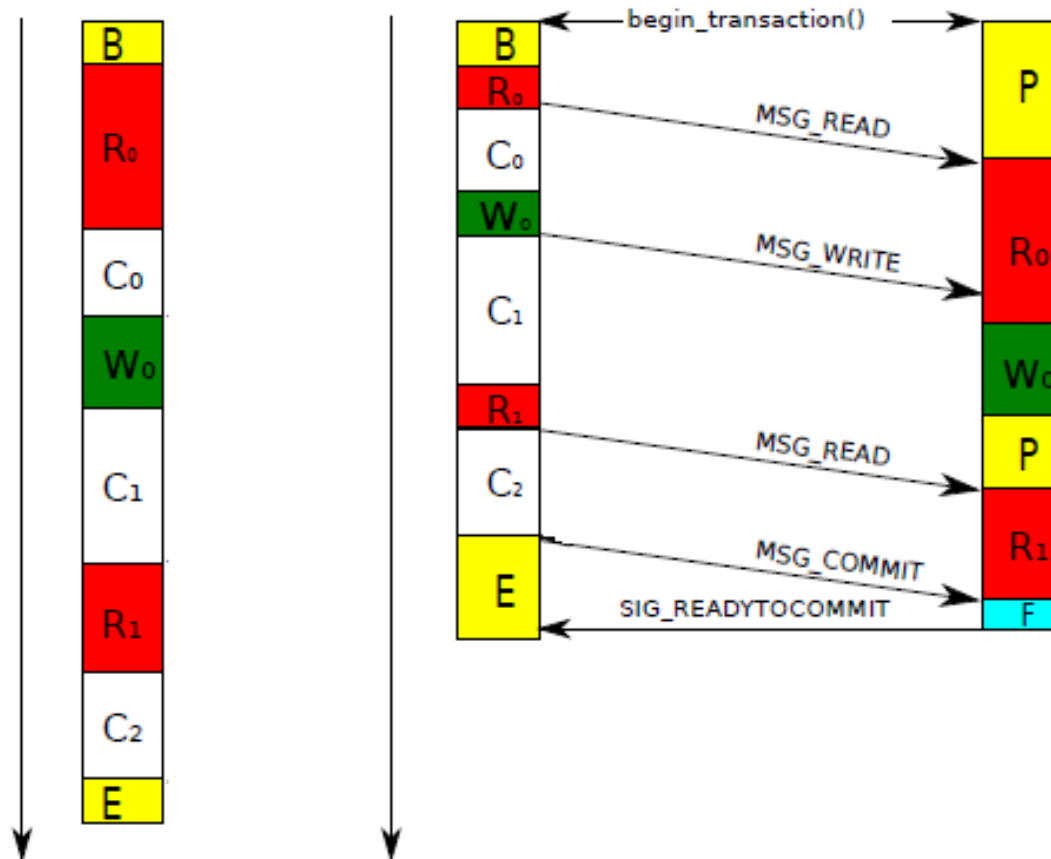


# Problems of TLSTM: Atomic operations on several tasks

- Leveraging STM with TLS features
  - Transaction validation is done separately by each task
  - Transaction validation was not needed in STM at commit time

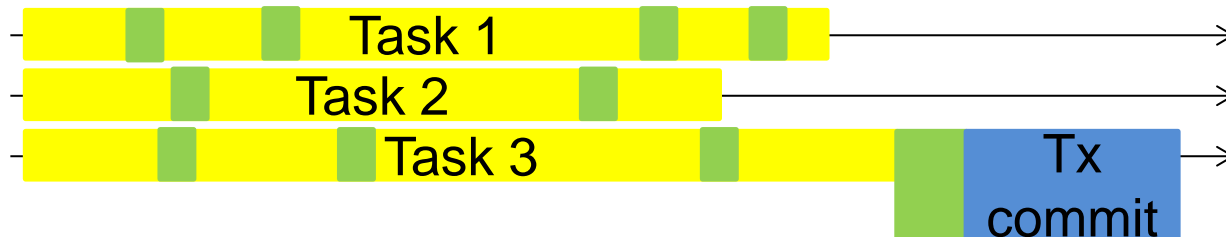


- Helper Threads [Kestor et al. 2011]

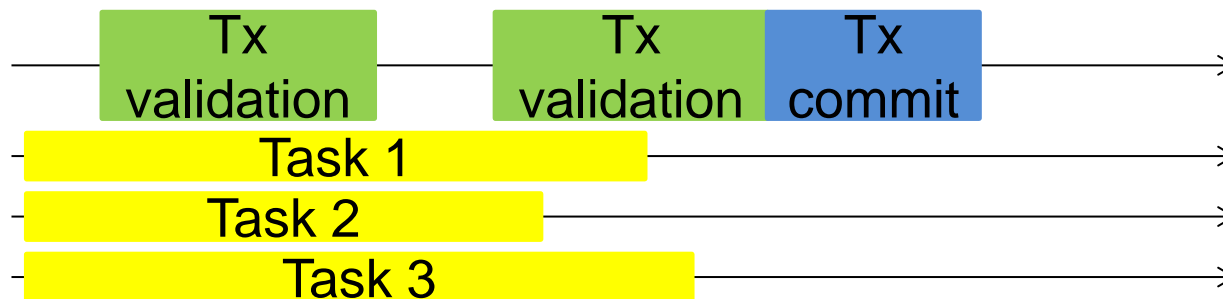


# Applying Helper Threads to TLS+STM: Solves the atomic Tx operations problem

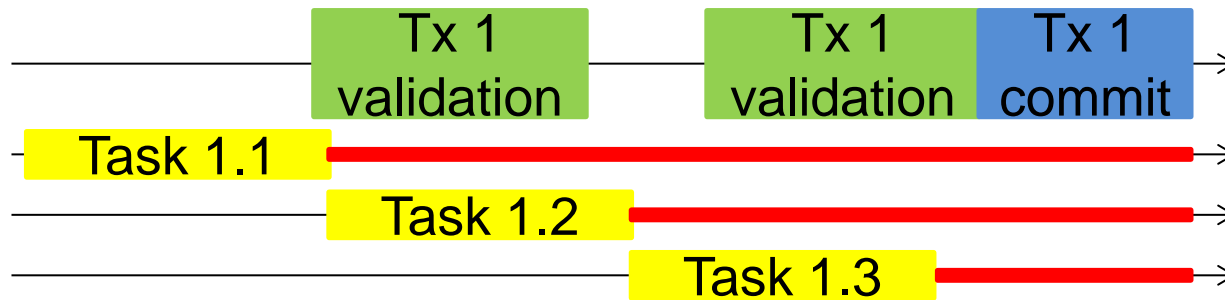
- Without Helper Threads



- With Helper Threads

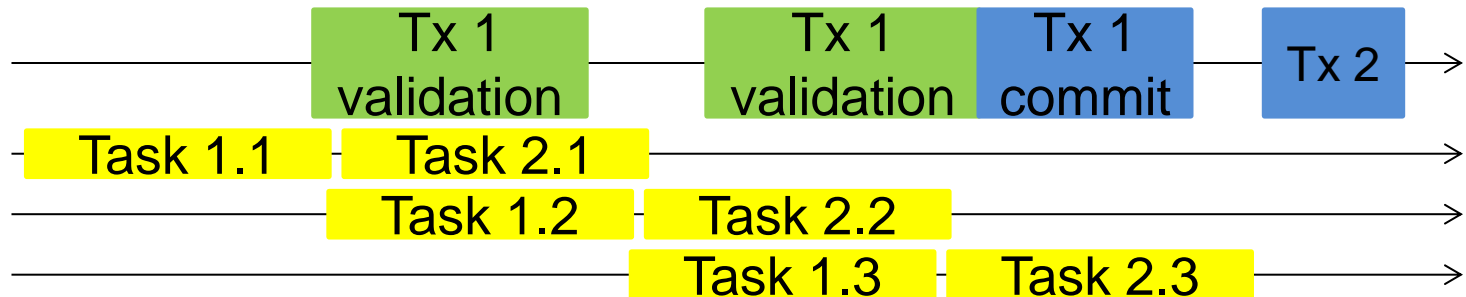
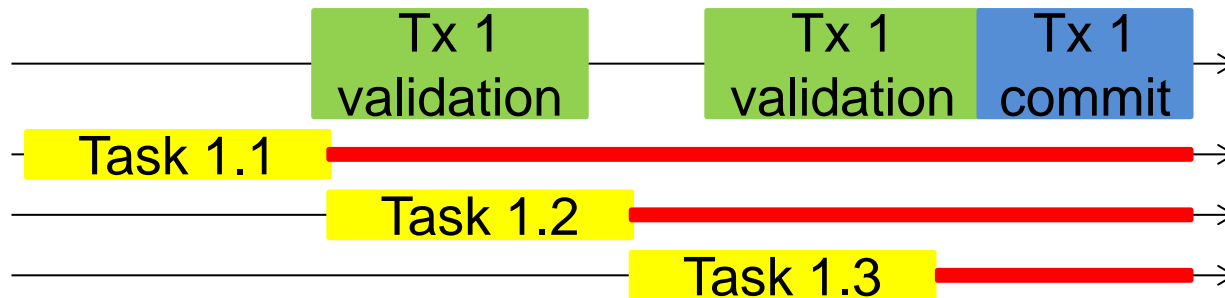


- Worst case scenario (all tasks conflict in the last instruction)



 Idle Time

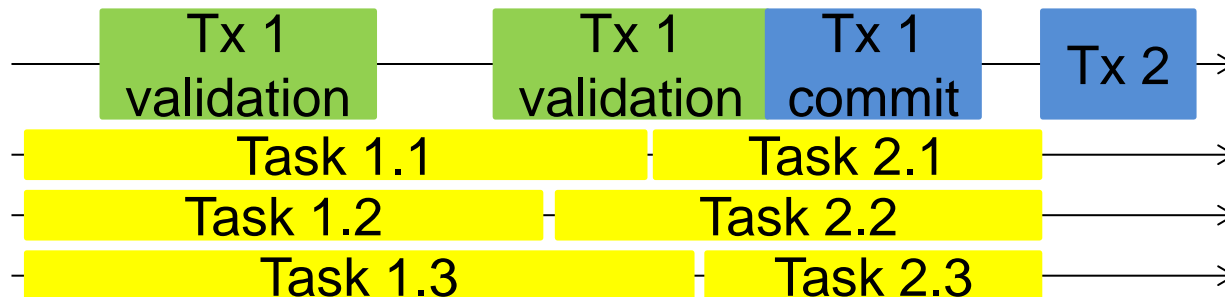
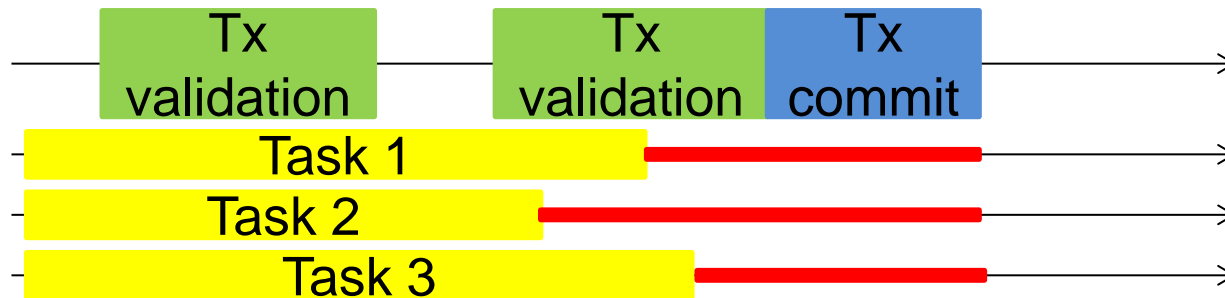
- Worst case scenario (all tasks conflict in the last instruction)



 Idle Time

# We can also eliminate idle time!

- Best case scenario (no task conflicts)



 Idle Time

- 10% overall improvement only by offloading transaction validation to helper threads (3 tasks per thread)



- 10% overall improvement only by offloading transaction validation to helper threads (3 tasks per thread)
- How many transactions can we run ahead of time?

- 10% overall improvement only by offloading transaction validation to helper threads (3 tasks per thread)
- How many transactions can we run ahead of time?
- How do Helper Threads affect cross-transaction conflicts?

- 10% overall improvement only by offloading transaction validation to helper threads (3 tasks per thread)
- How many transactions can we run ahead of time?
- How do Helper Threads affect cross-transaction conflicts?
- Opens room to use TLS+STM with smaller tasks?

- 10% overall improvement only by offloading transaction validation to helper threads (3 tasks per thread)
- How many transactions can we run ahead of time?
- How do Helper Threads affect cross-transaction conflicts?
- Opens room to use TLS+STM with smaller tasks?
- We **NEED** more benchmarks

# Questions?

**technology  
from seed**



<http://www.gsd.inesc-id.pt/project-pages/specSTM>