

# FASTLANE

## Streamlining Transactions for Low Thread Counts

*Jons-Tobias Wamhoff*

Christof Fetzer

Technische Universität Dresden, Germany

Pascal Felber

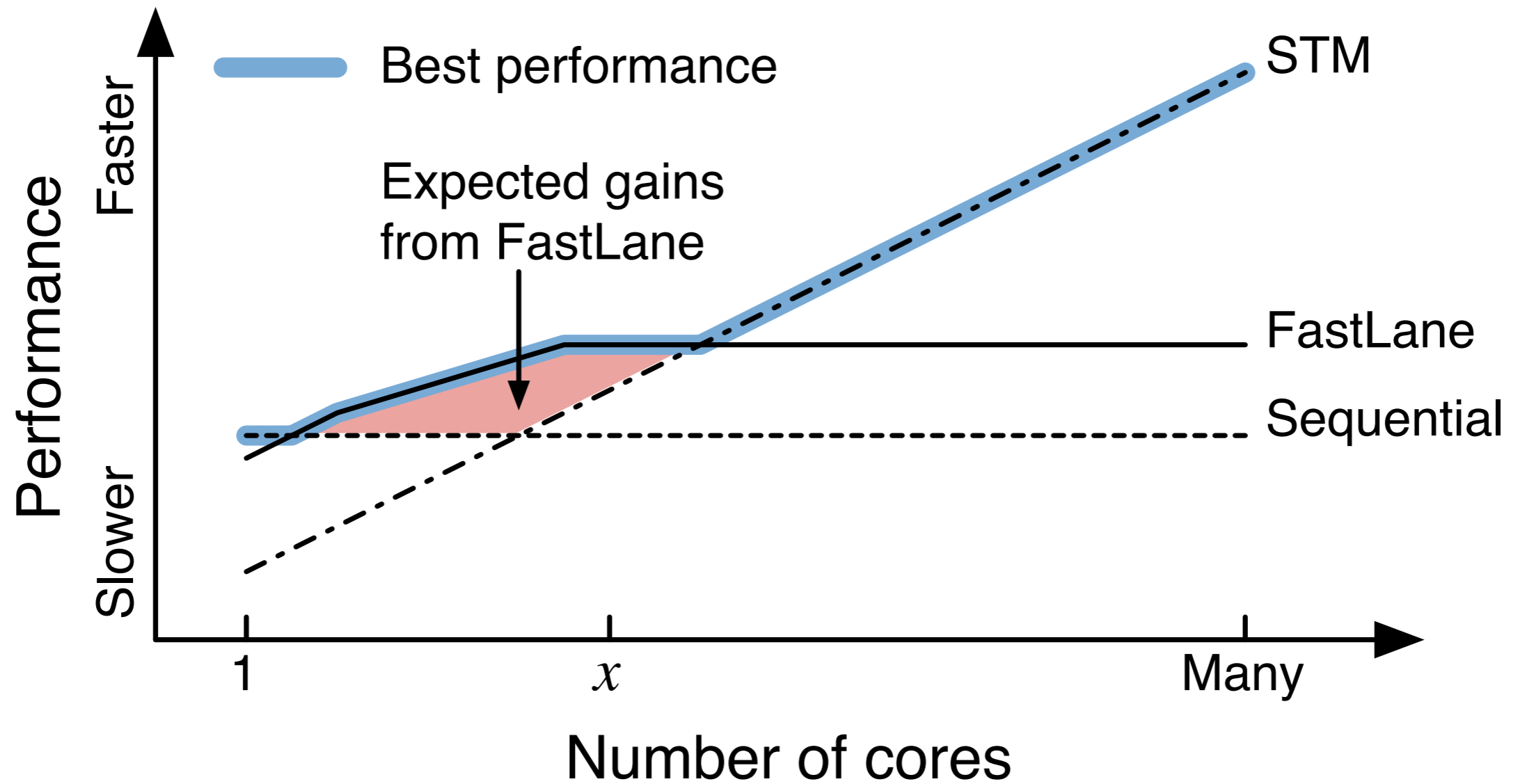
Etienne Rivière

Université de Neuchâtel, Switzerland

Gilles Muller

INRIA, France

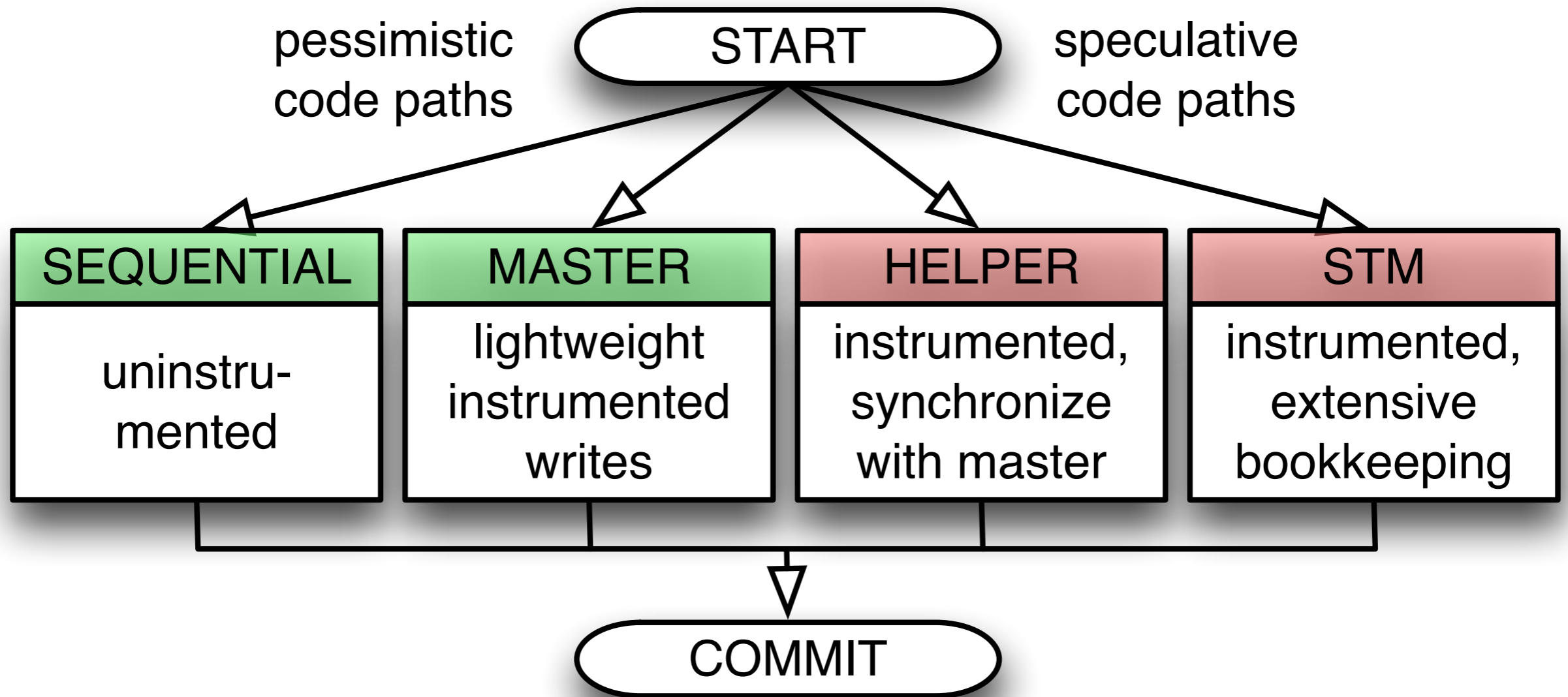
# Motivation



# General Idea

- 1 master thread
  - Commits transactions without aborting
  - Minimal instrumentation and bookkeeping
- N helper threads
  - Commit transactions only when not in conflict
  - Contribute progress without impairing on the performance of the master

# Code Paths

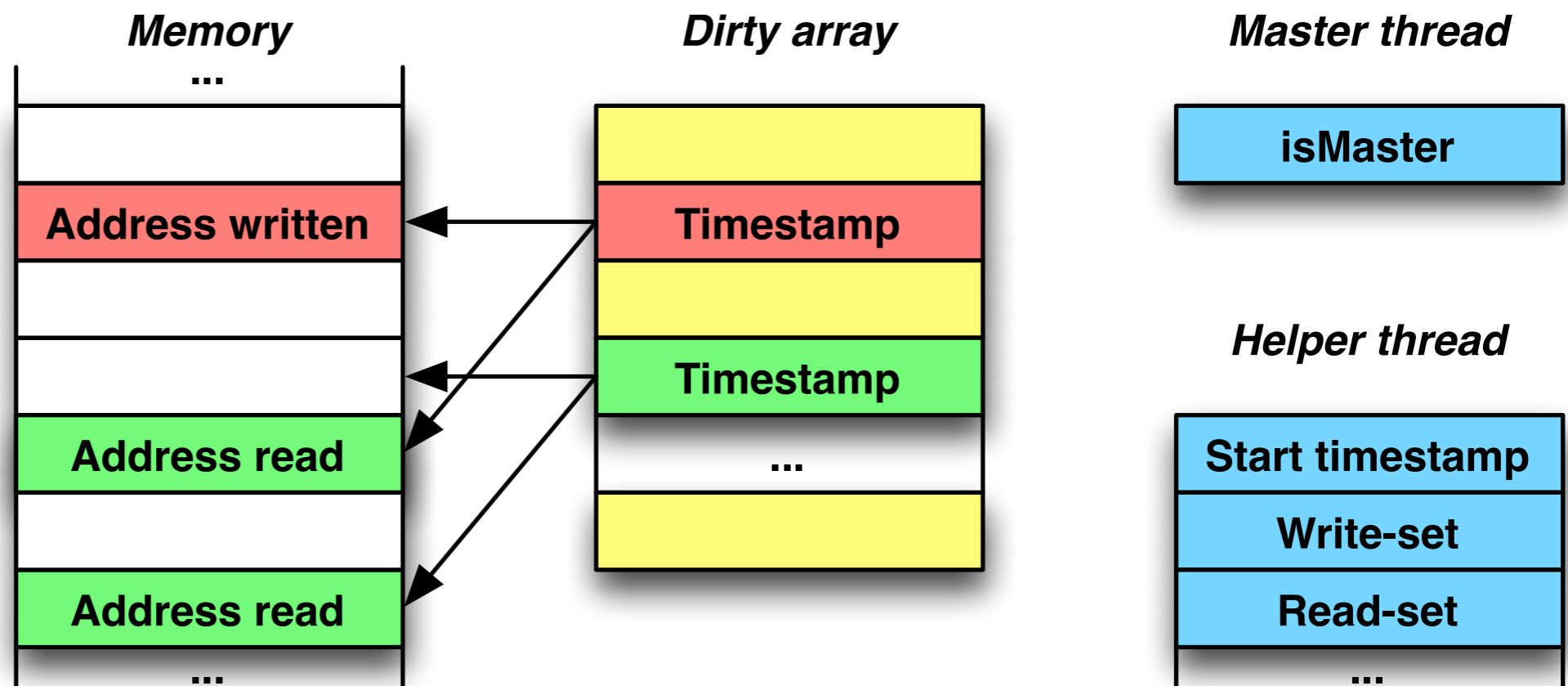


# Code Paths

- Dresden TM Compiler
  - Generates multiple code paths for sequential (uninstrumented), FastLane (master & helper) and STM
  - Generic `START` and `COMMIT` calls with internal branch
  - `READ` and `WRITE` are specific to code path and inlined
    - transaction descriptor only accessed if needed
- TinySTM++ TM runtime
  - Dynamically select code path based on core or thread count at `BEGIN`

# Data Structures

**Counter**  
odd: owned  
even: otherwise



# Master vs. Helper

MASTER

BEGIN

acquire(*cntr*)

READ (*addr*)

return \**addr*

WRITE (*addr*, *val*)

*dirty*[hash(*addr*)] = *cntr*

*addr* = *val*

COMMIT

release(*cntr*)

HELPER

BEGIN

start = *cntr*

READ (*addr*)

*dirty*[hash(*addr*)] ≤ start

add(read-set, *addr*)

return \**addr*

abort  
→

WRITE (*addr*, *val*)

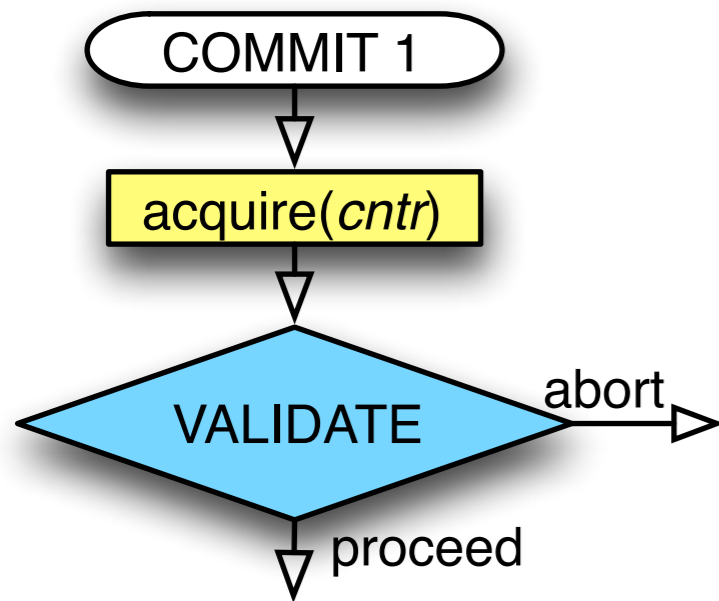
*dirty*[hash(*addr*)] ≤ start

put(write-set, *addr*, *val*)

abort  
→

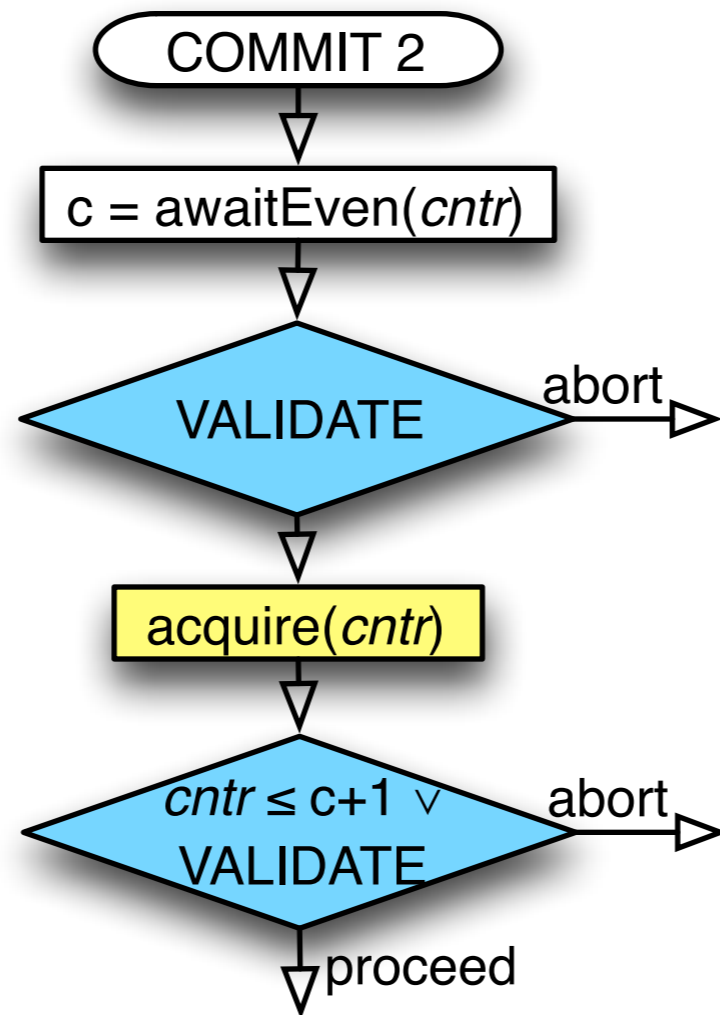
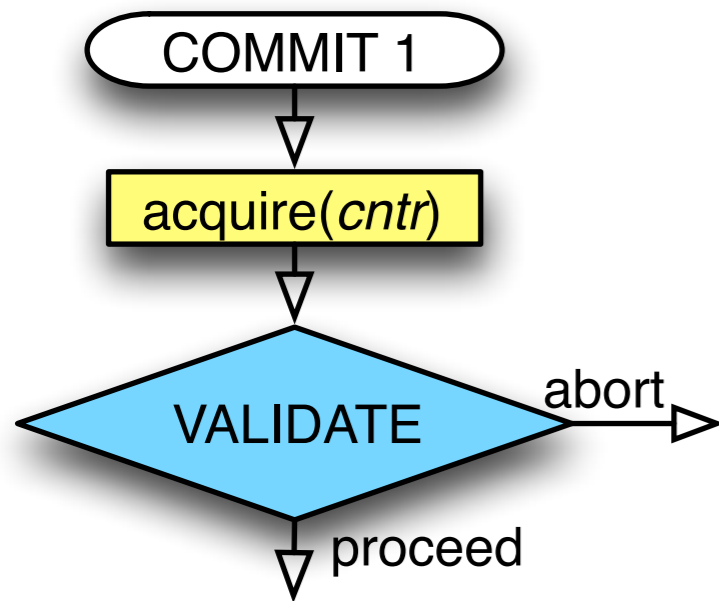
COMMIT

# 3 Commit Variants

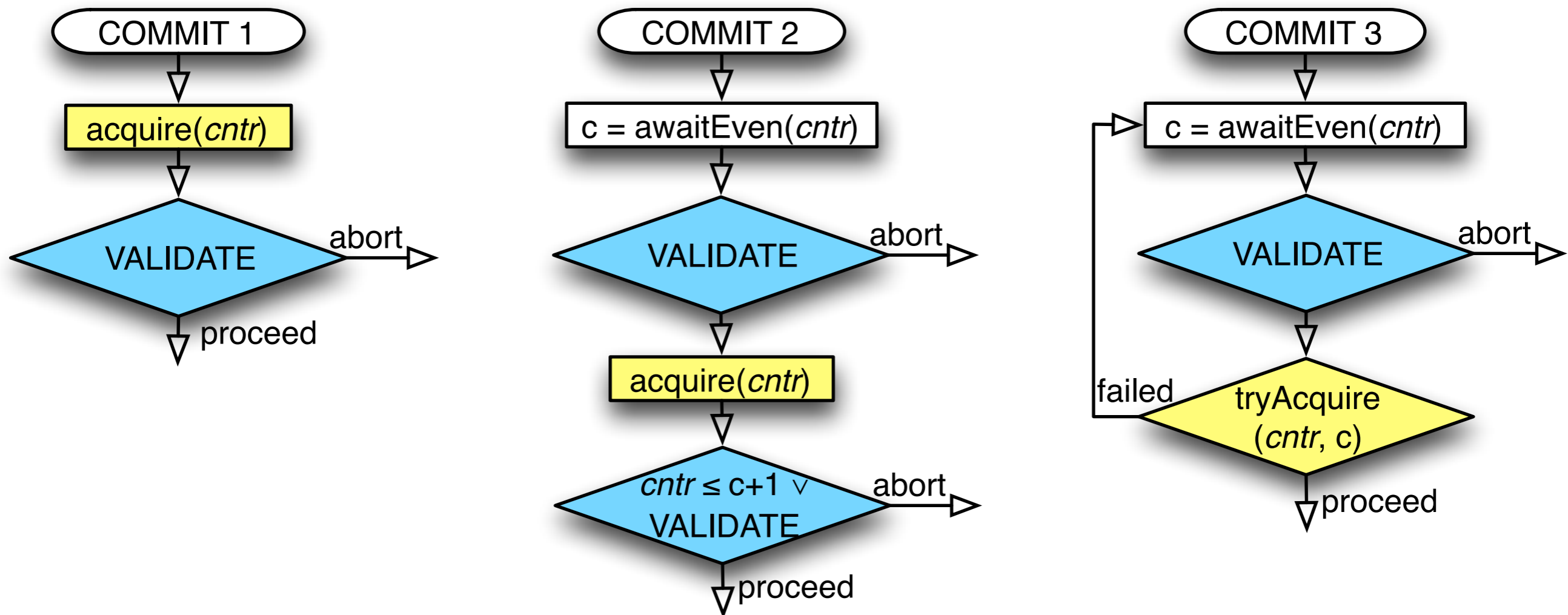




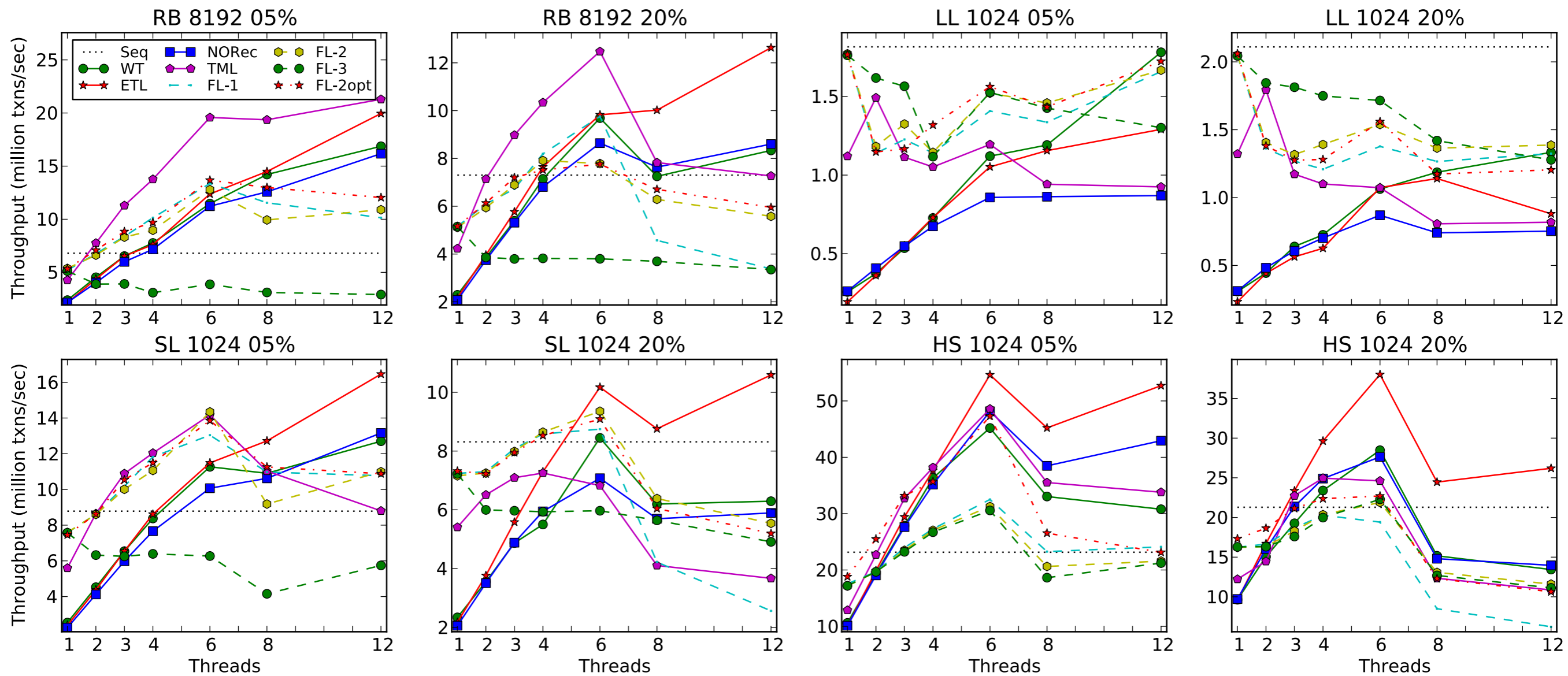
# 3 Commit Variants



# 3 Commit Variants



# Intset Benchmarks



**Thank you!**