Effective Barrier Synchronization on Intel Xeon Phi Coprocessor

Andrey Rodchenko, Andy Nisbet, Antoniu Pop, Mikel Lujan

Advanced Processor Technologies Group,
School Of Computer Science,
The University of Manchester

Open-source: https://github.com/arodchen/cbarriers

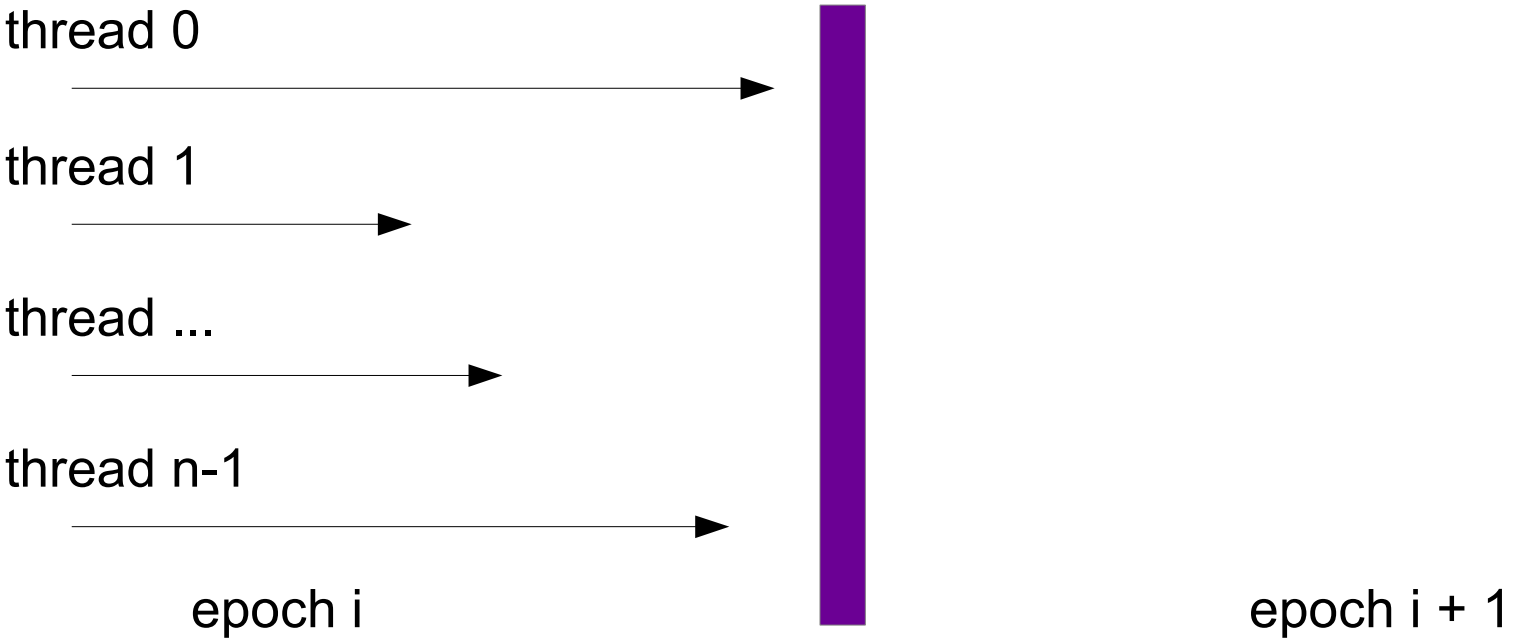
Overview

MANCHESTER 1824
The University of Manchester

- What is barrier synchronization?
- Intel Xeon Phi Coprocessor Architecture
- Review of Barrier Synchronization Algorithms
- Novel Hybrid Barrier Synchronization Algorithm
- Evaluation of Barrier Synchronization Algorithms on Intel Xeon Phi Coprocessor
- Conclusions

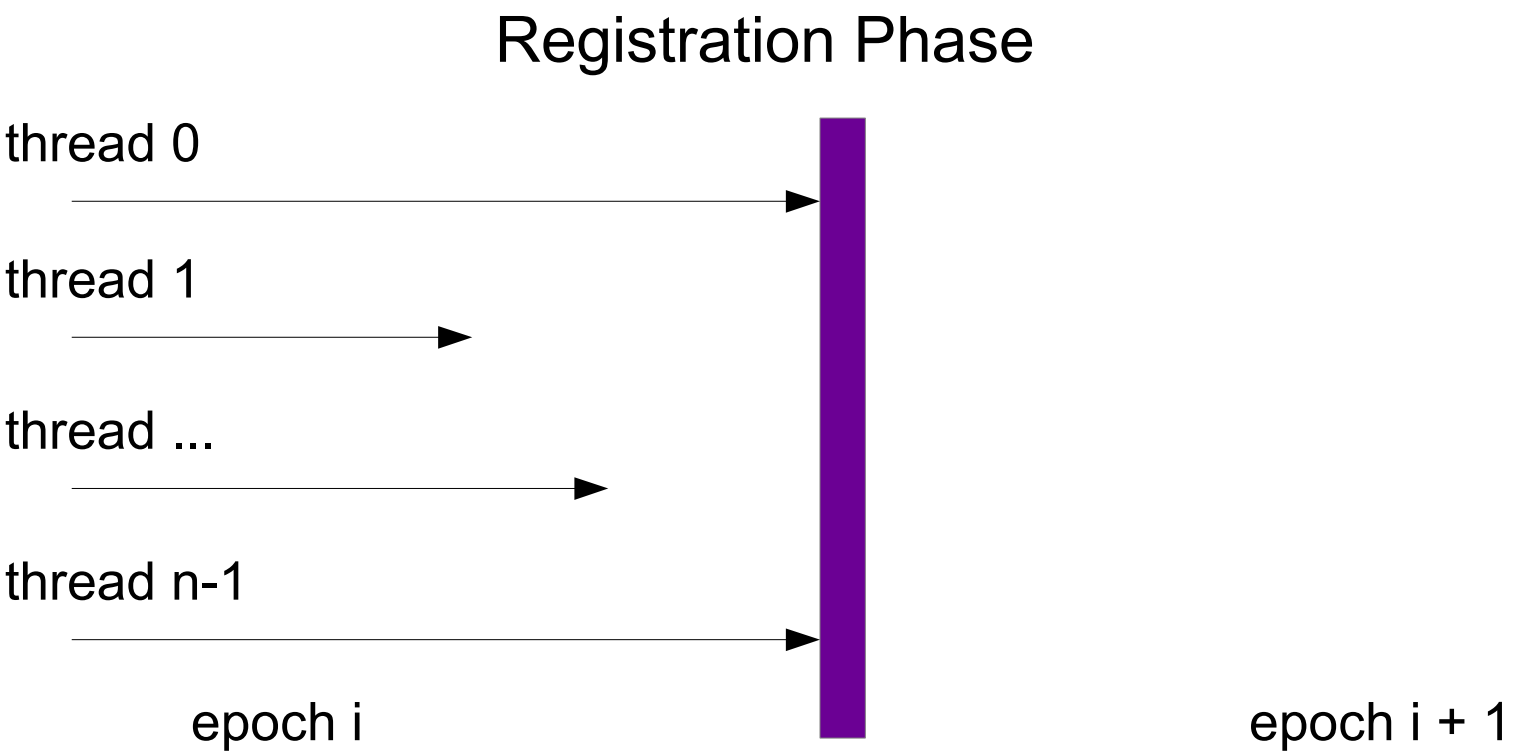
What is barrier synchronisation?

MANCHESTER 1824
The University of Manchester



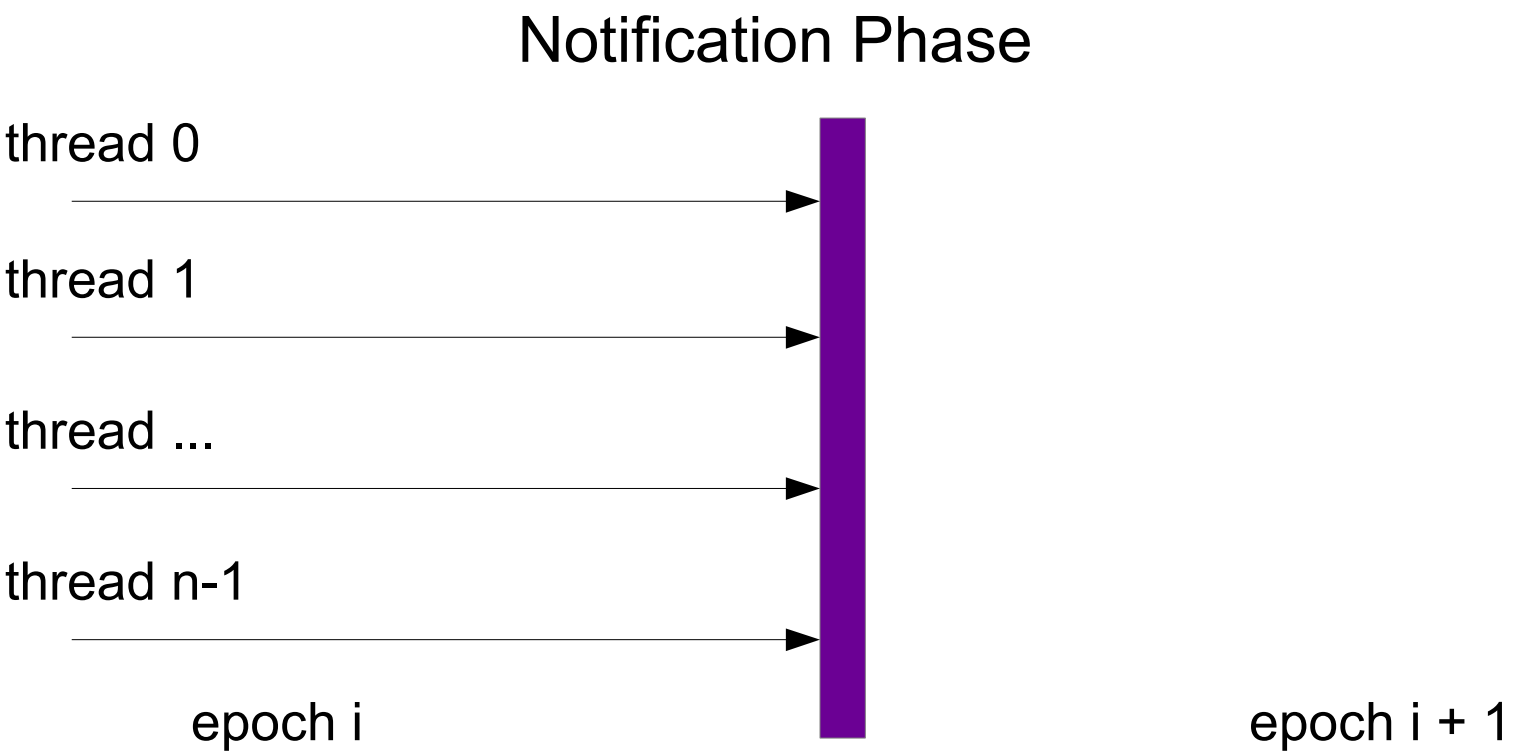
What is barrier synchronisation?

MANCHESTER 1824
The University of Manchester



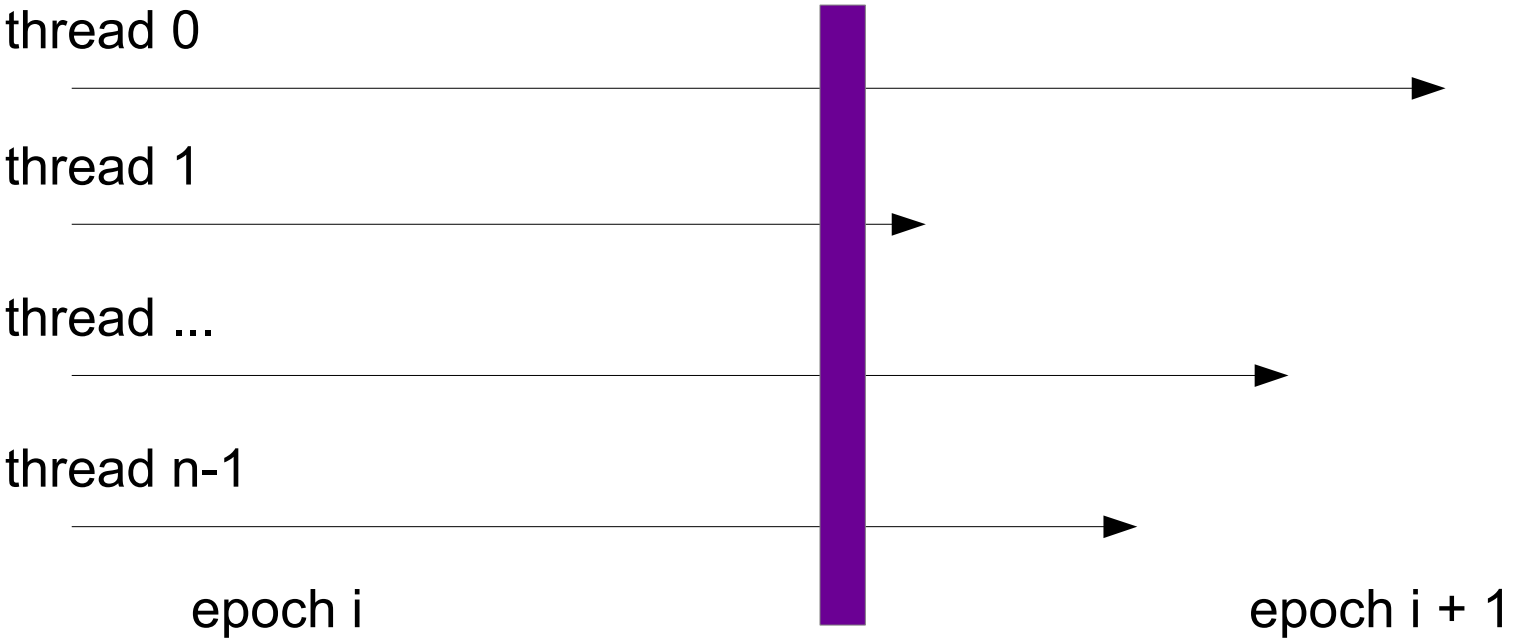
What is barrier synchronisation?

The University of Manchester



What is barrier synchronisation?

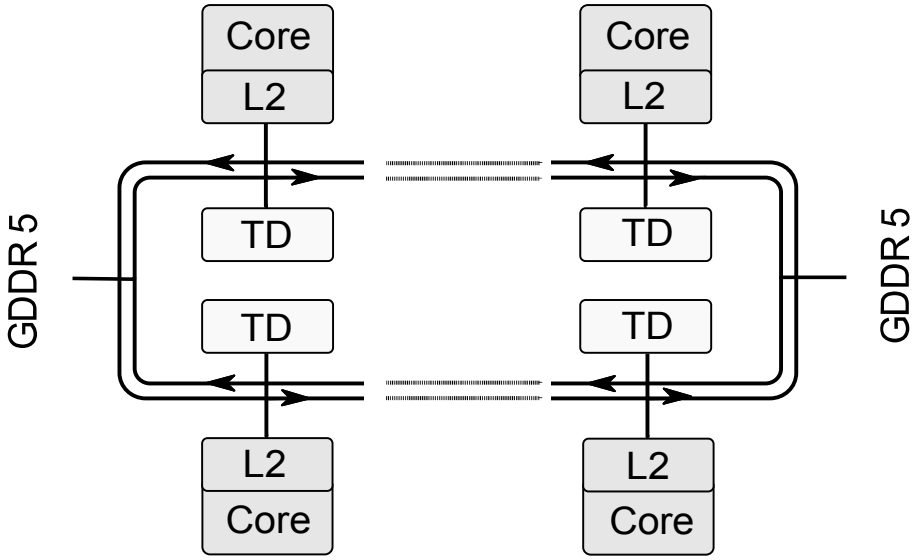
MANCHESTER 1824
The University of Manchester



Intel Xeon Phi Coprocessor

MANCHESTER 1824
The University of Manchester

- Xeon Phi 5110P (Knights Corner)
  - 60 cores
  - cores are in order 4 way SMT
  - bidirectional ring interconnect



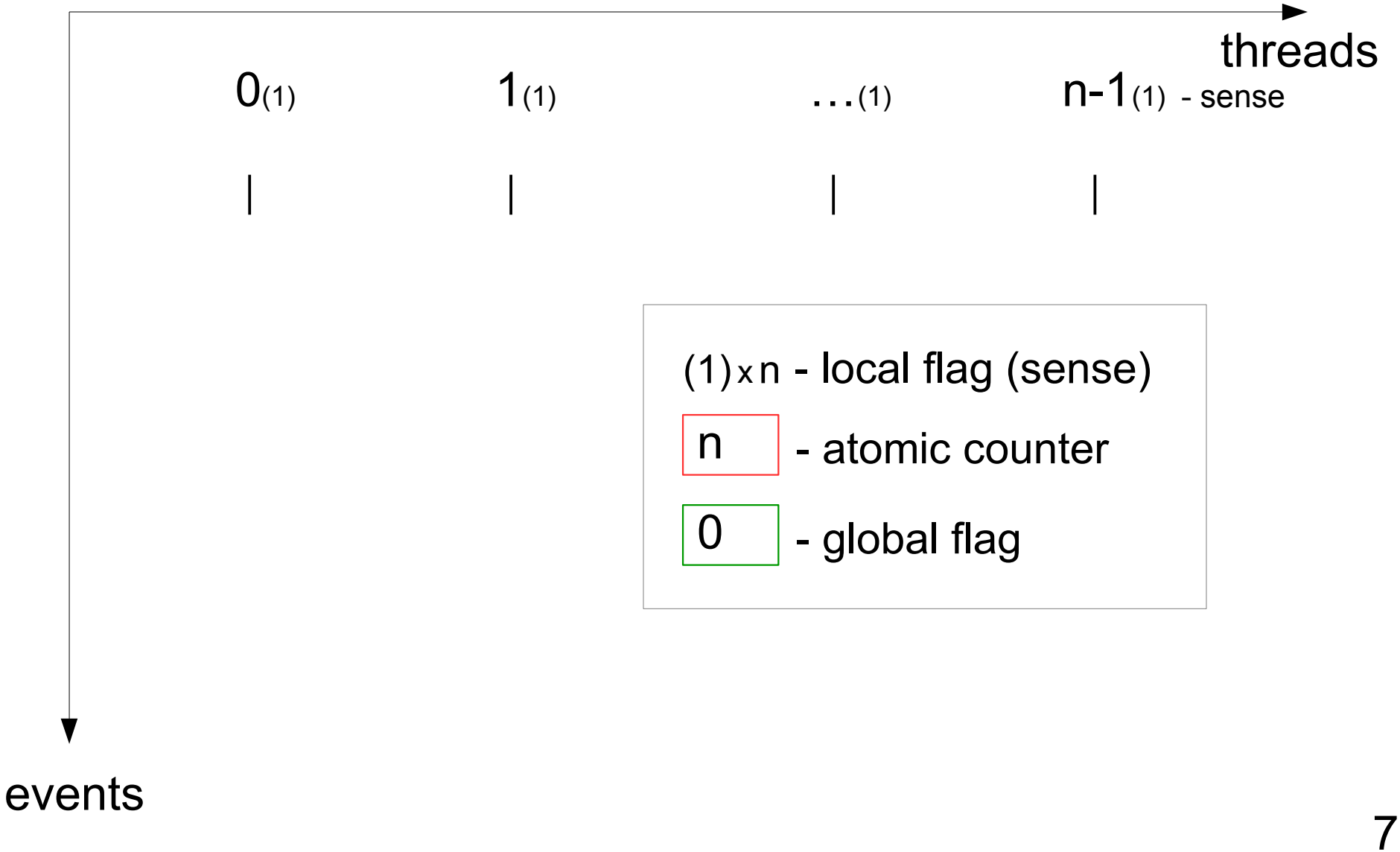
- Cache Coherence
  - extended MESI protocol
  - S state is extended with GOLS protocol via tag directories (TD)

- Specific Memory Access Instructions
  - globally ordered streaming stores: `vmovnrap[d/s]`
  - non-globally ordered streaming stores: `vmovnrngoap[d/s]`

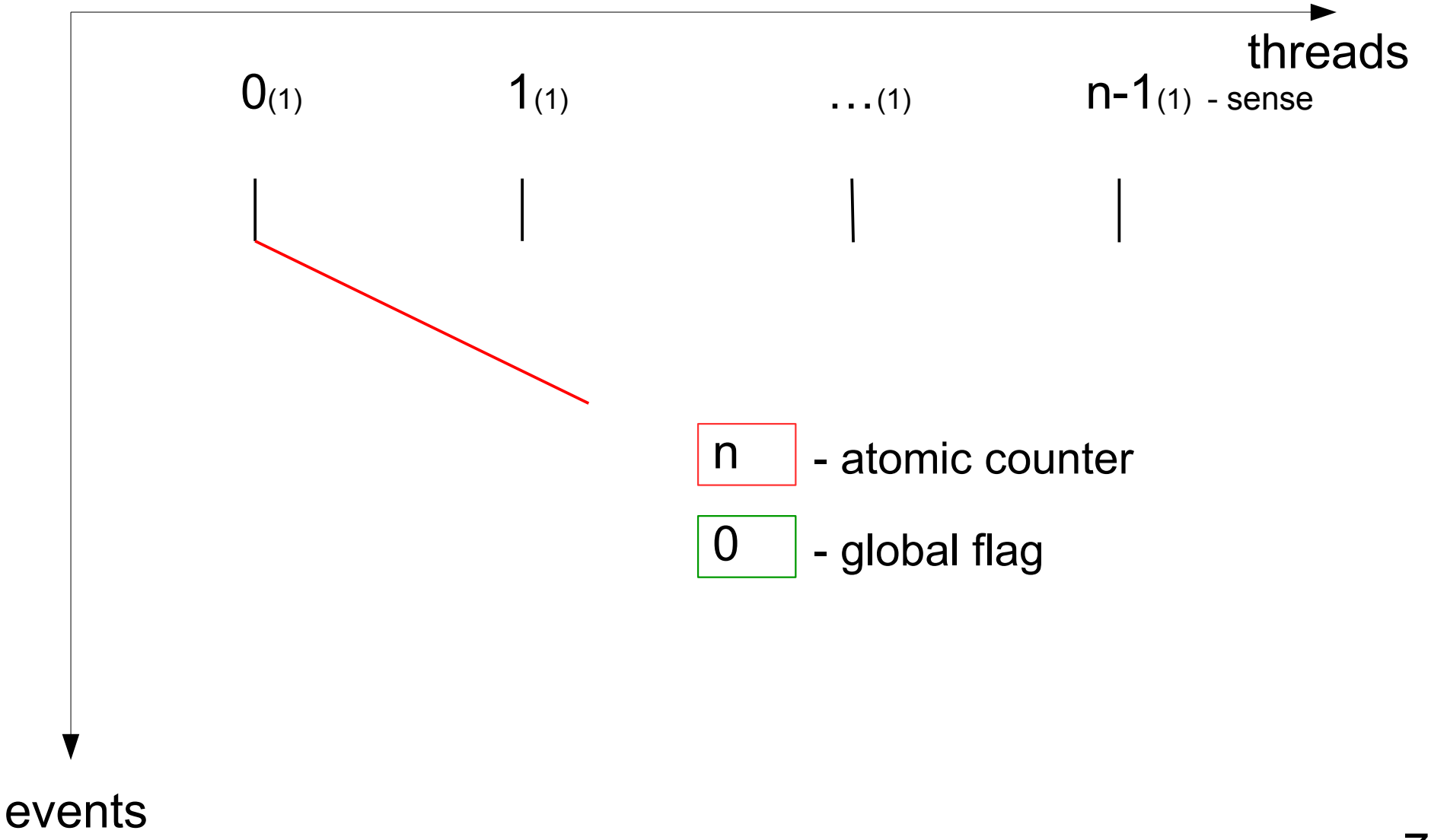
- Delay Instruction: `delay r32/r64`

Centralized Barrier Algorithm

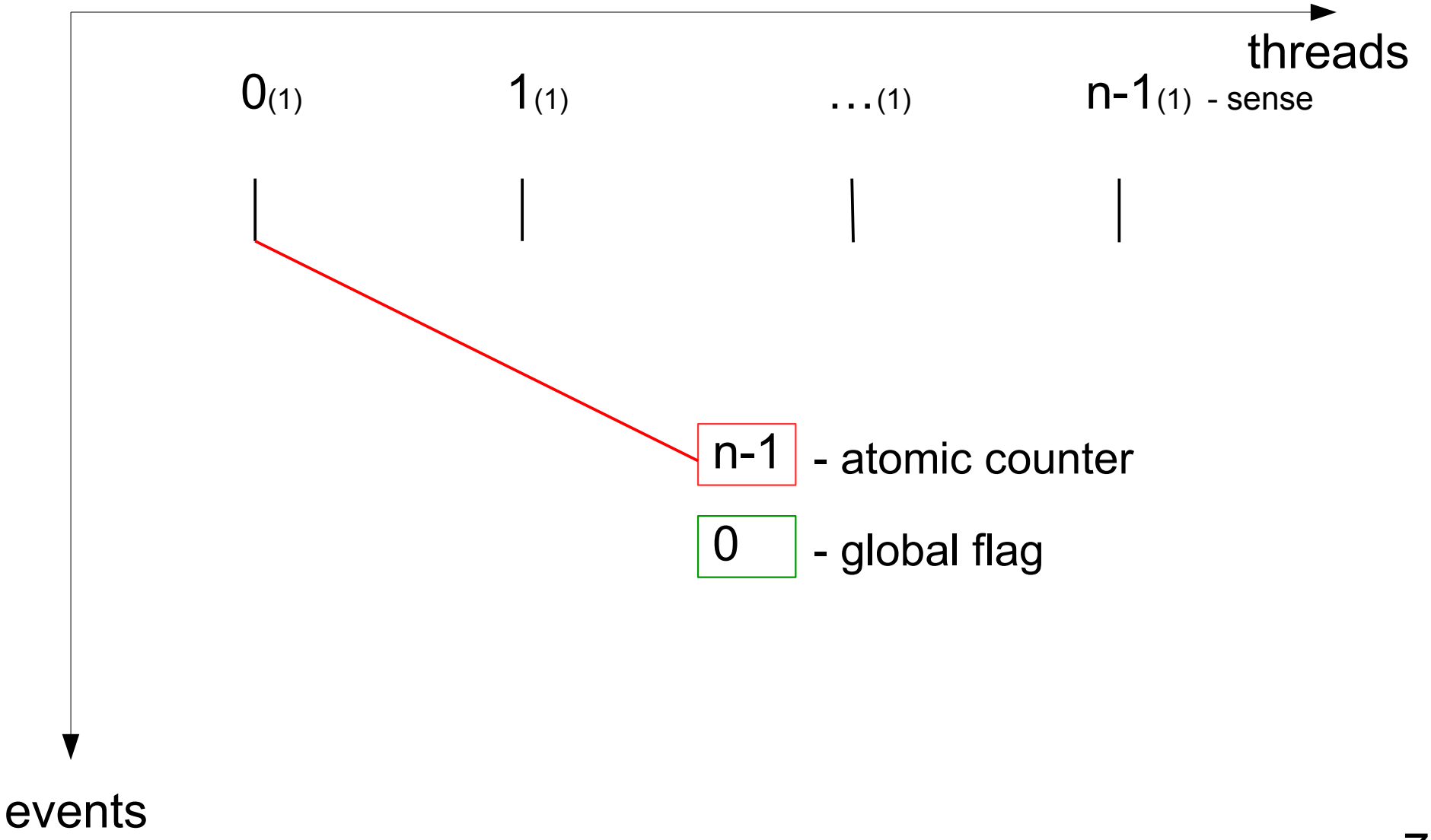
MANCHESTER 1824
The University of Manchester



Centralized Barrier Algorithm

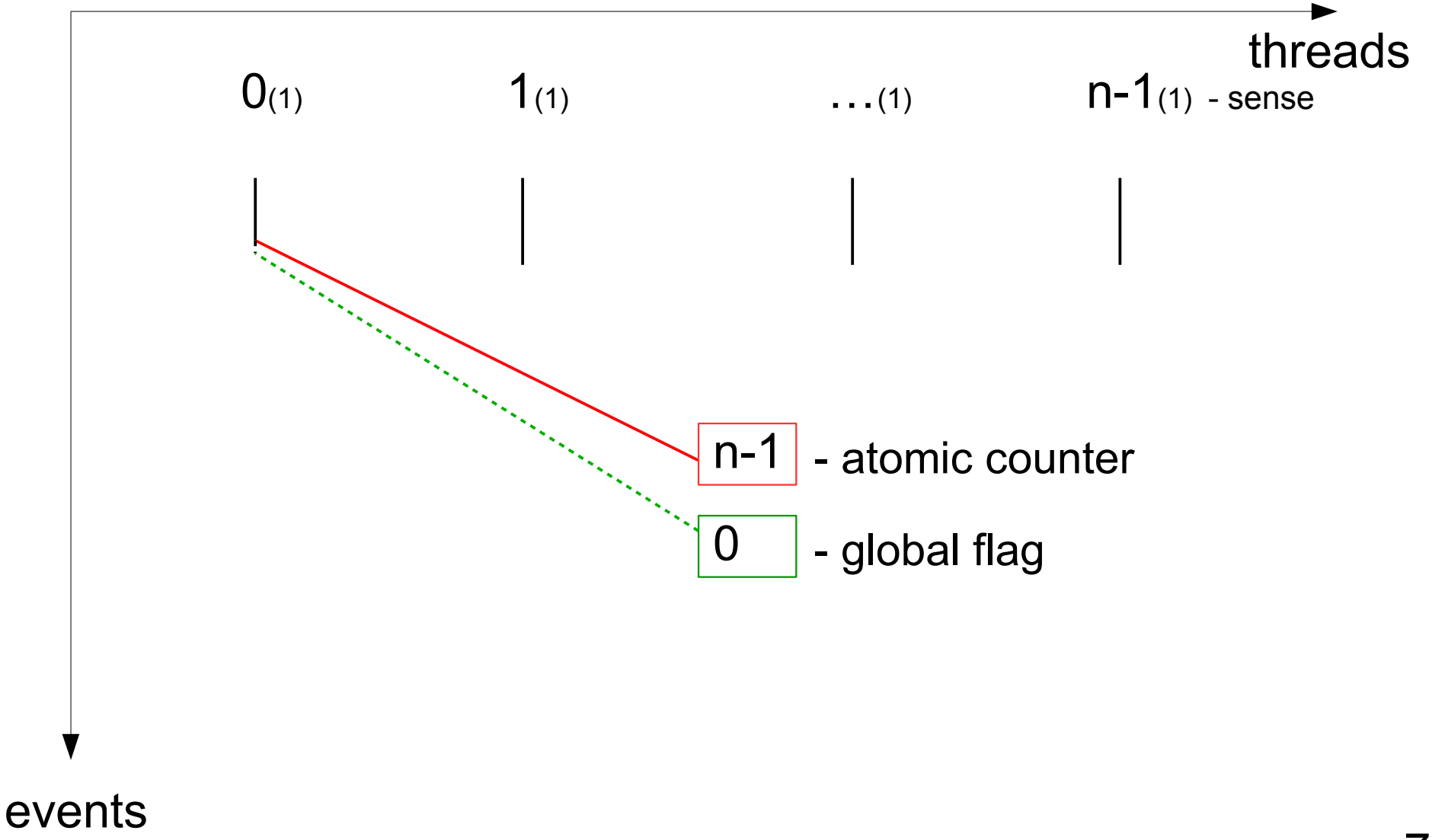


Centralized Barrier Algorithm

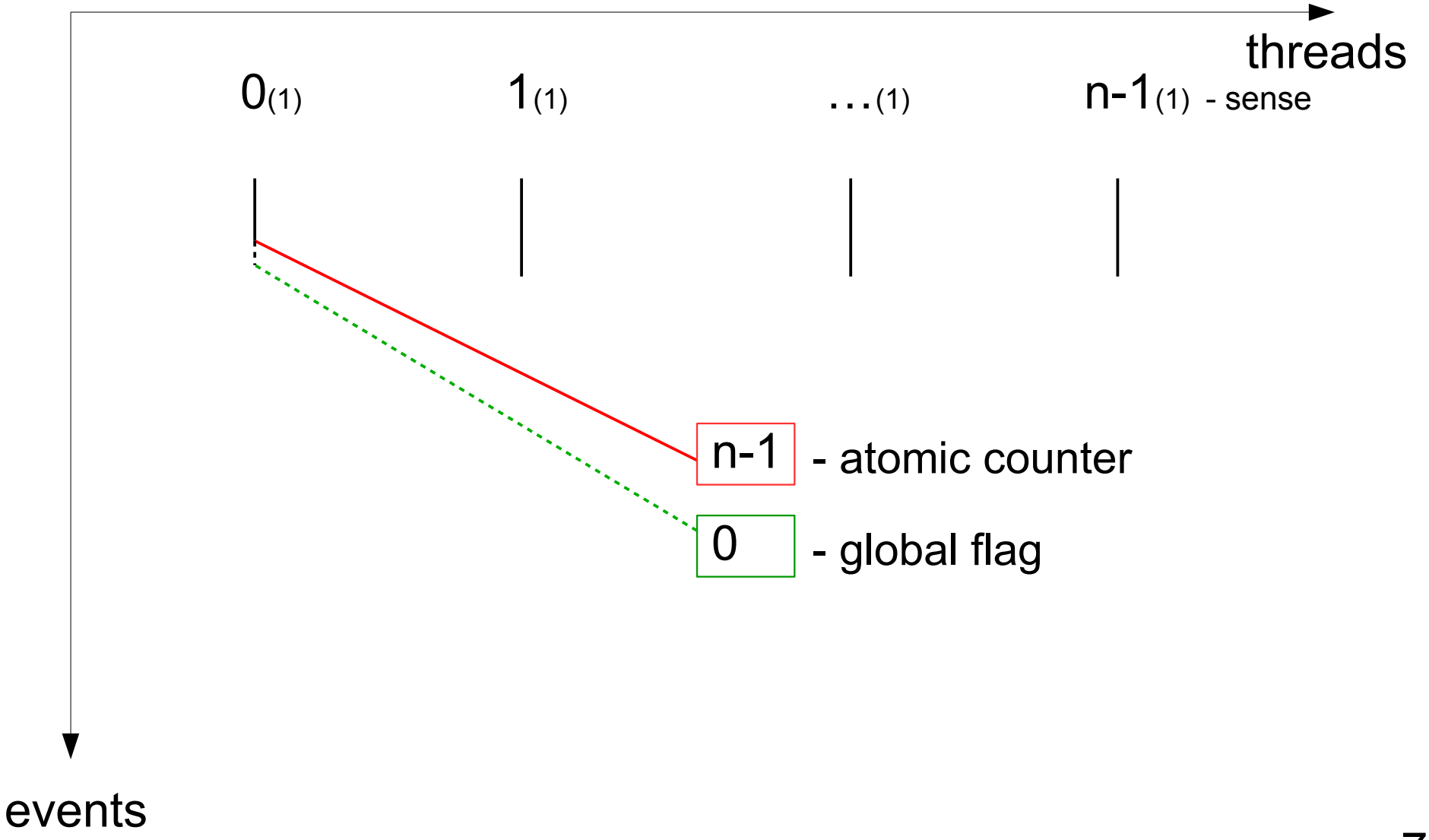


Centralized Barrier Algorithm

MANCHESTER 1824
The University of Manchester

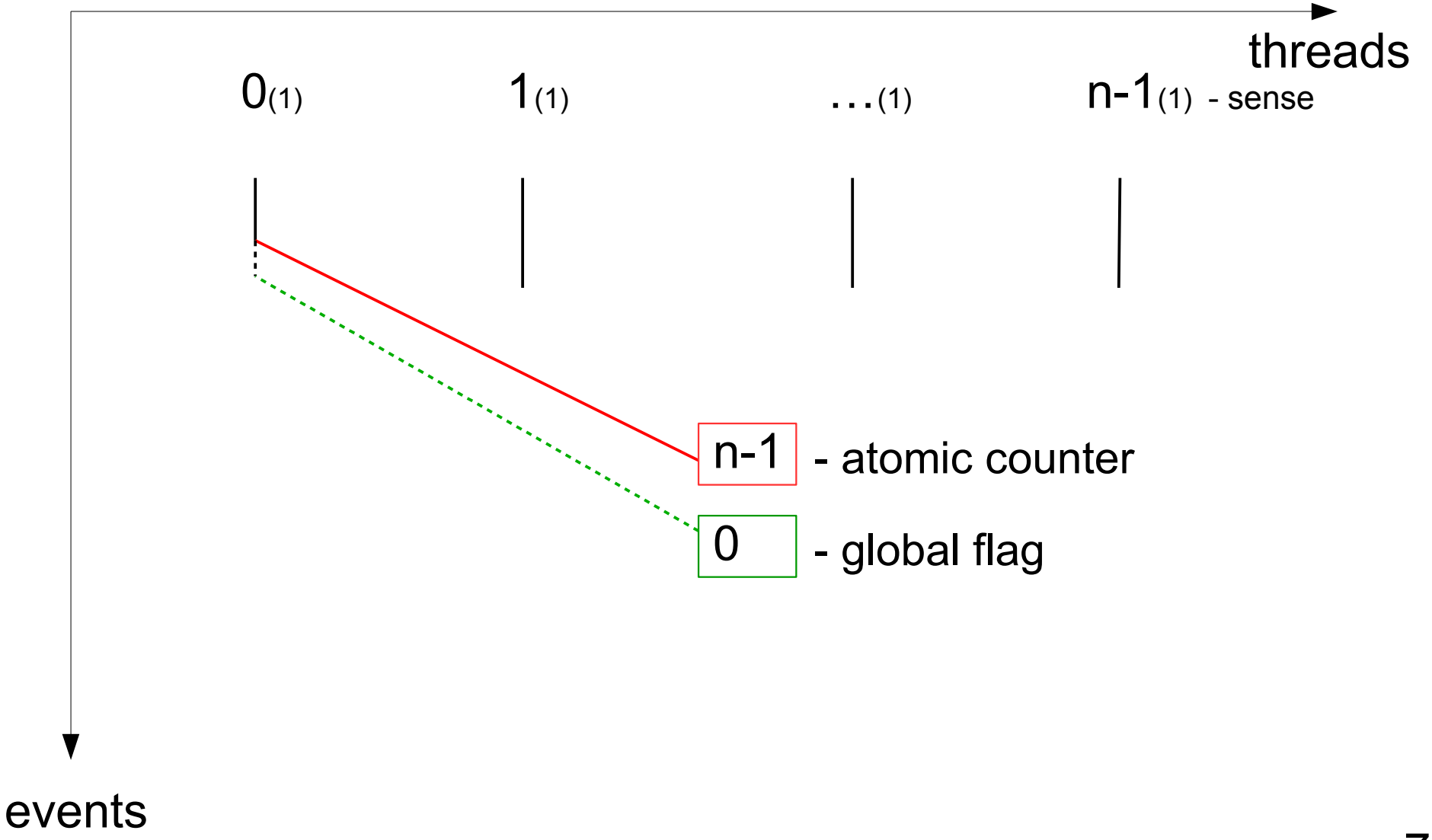


Centralized Barrier Algorithm



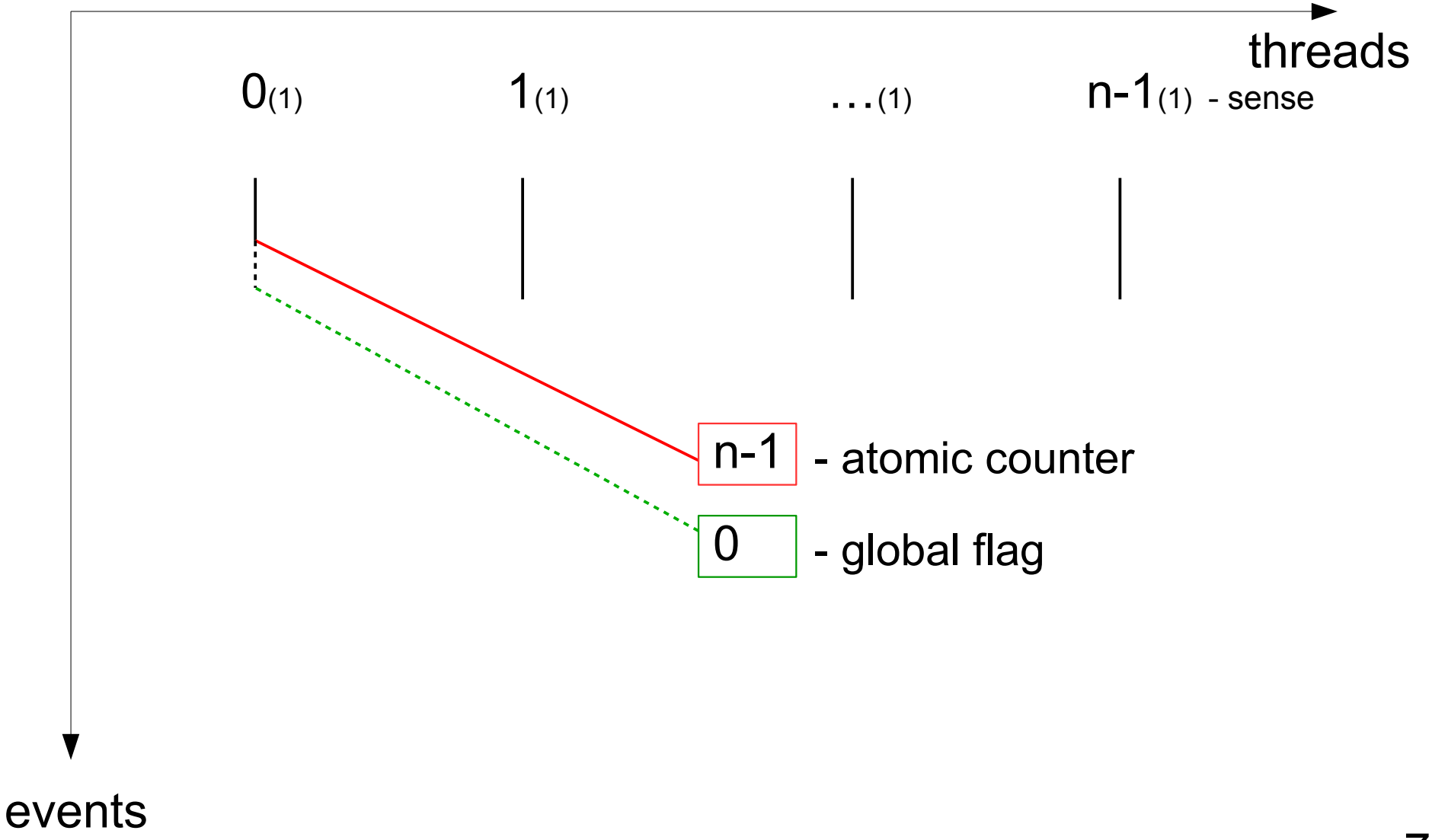
Centralized Barrier Algorithm

MANCHESTER 1824
The University of Manchester



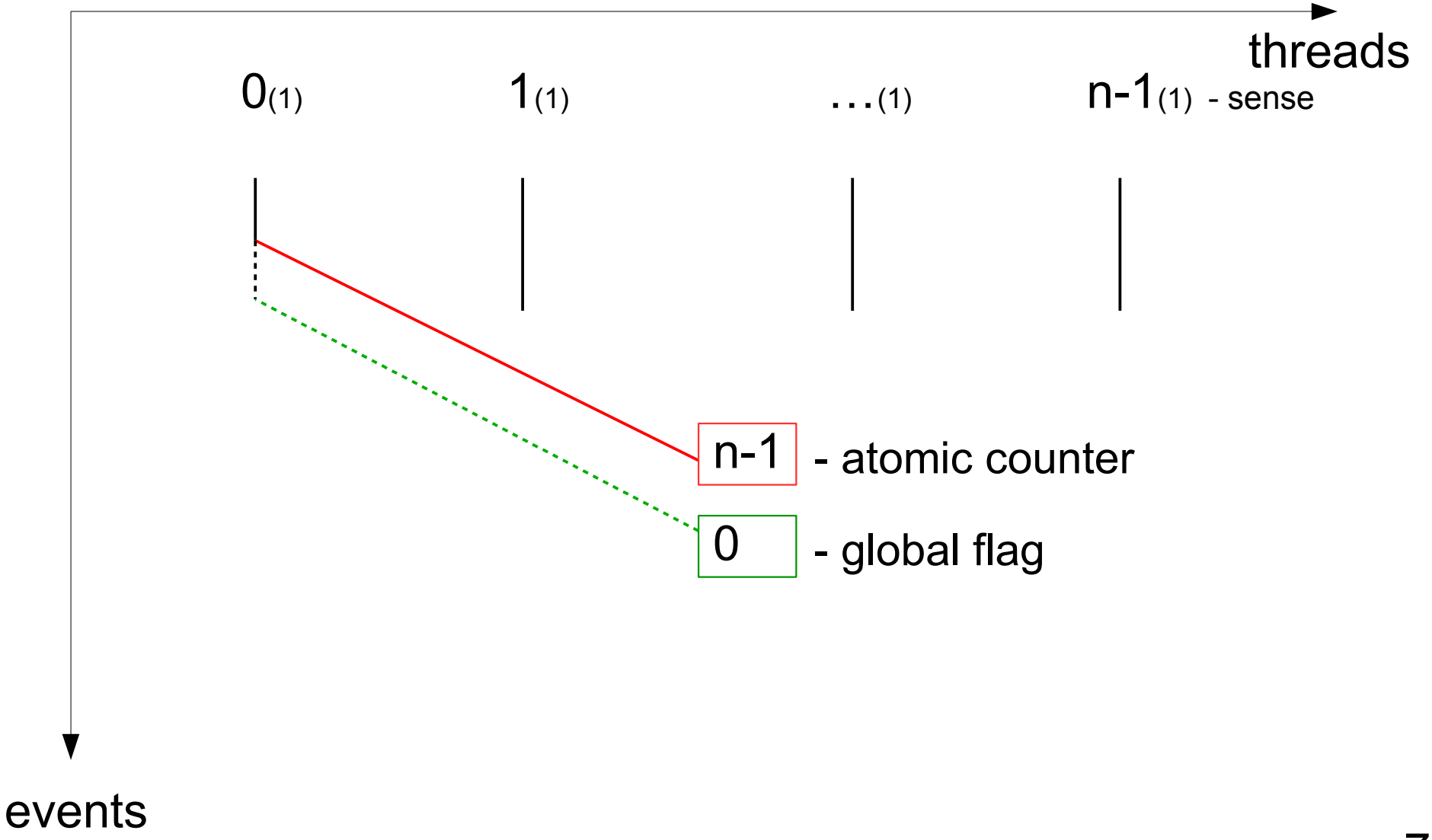
Centralized Barrier Algorithm

MANCHESTER 1824
The University of Manchester

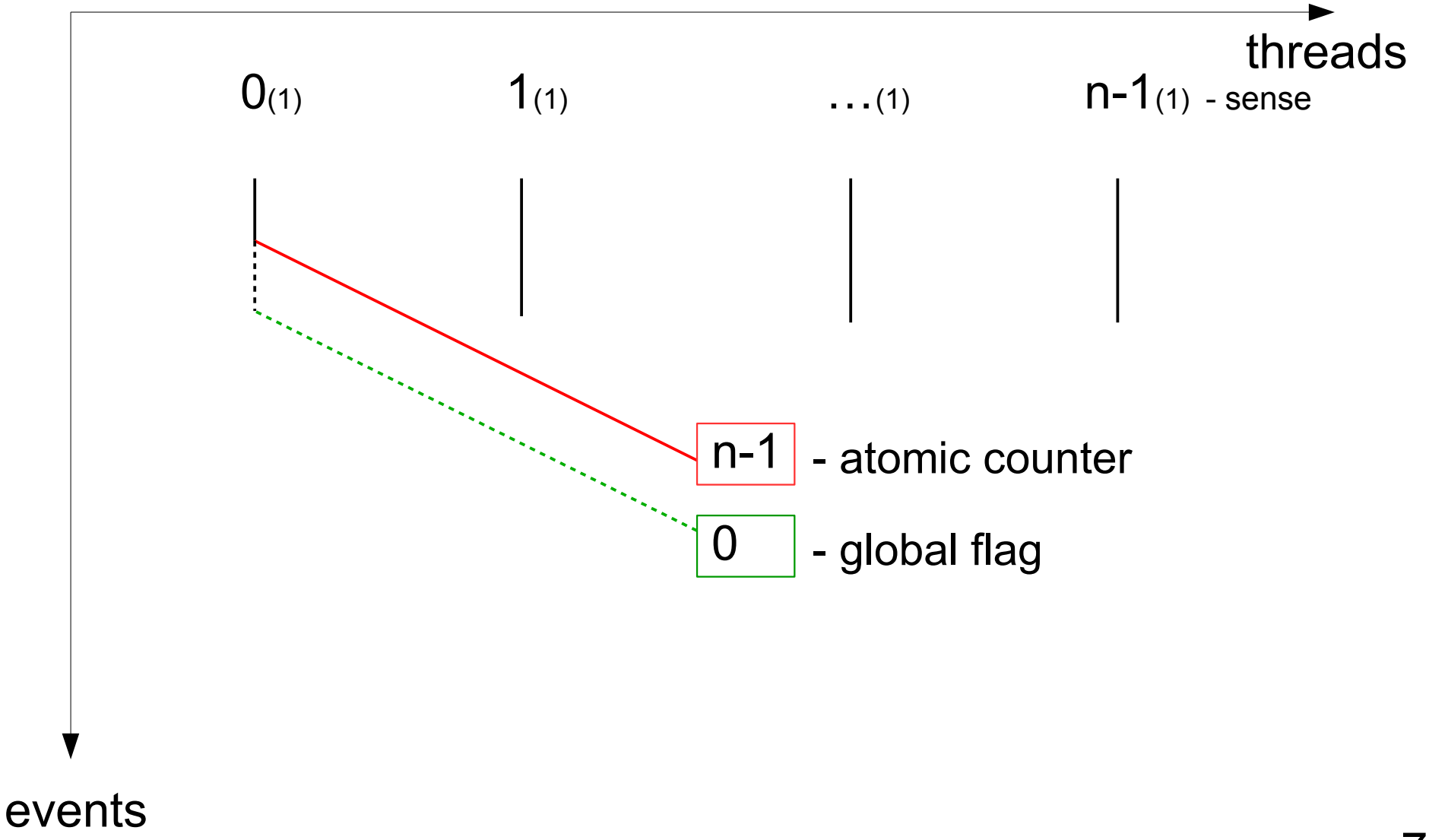


Centralized Barrier Algorithm

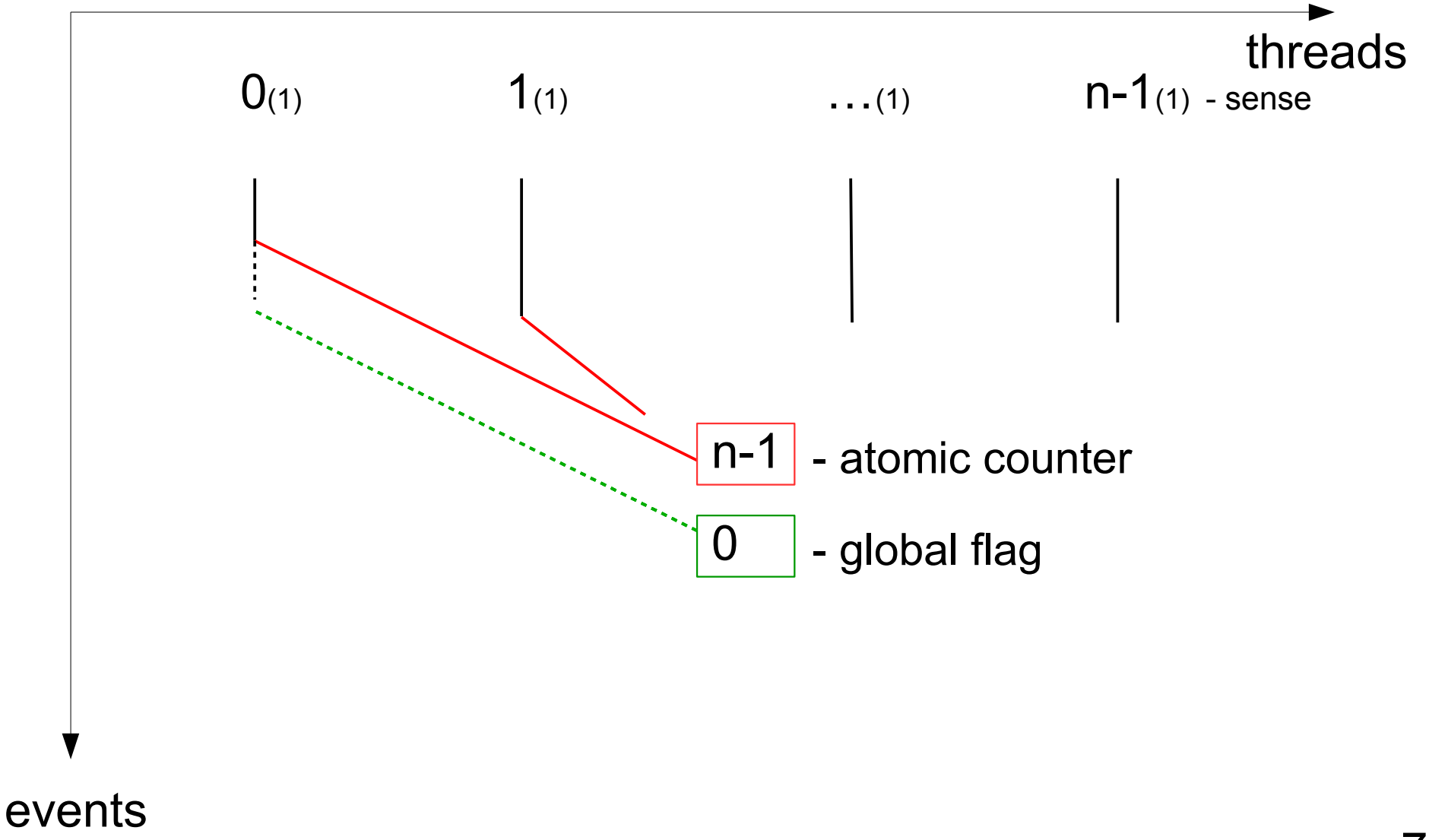
MANCHESTER 1824
The University of Manchester



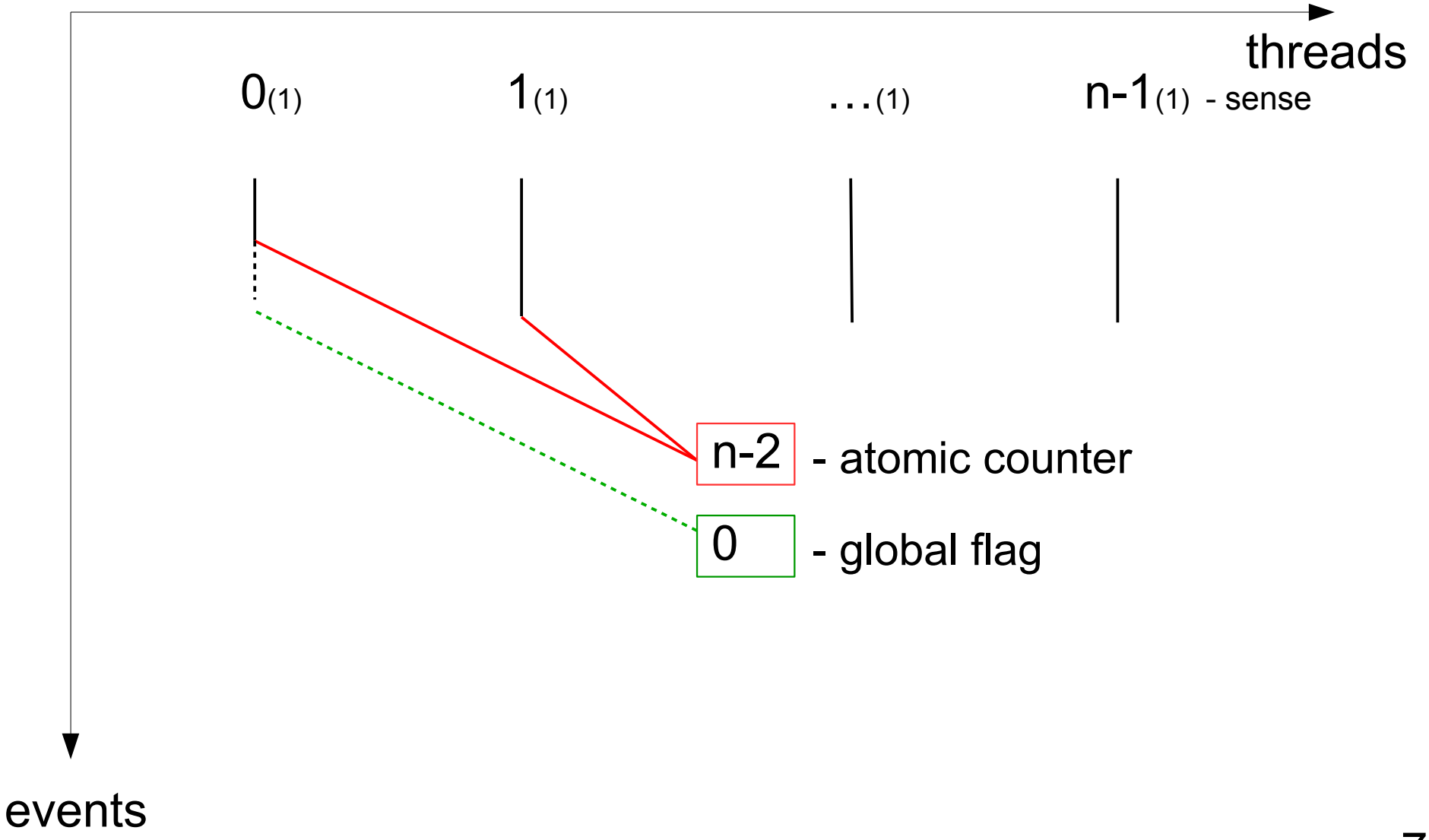
Centralized Barrier Algorithm



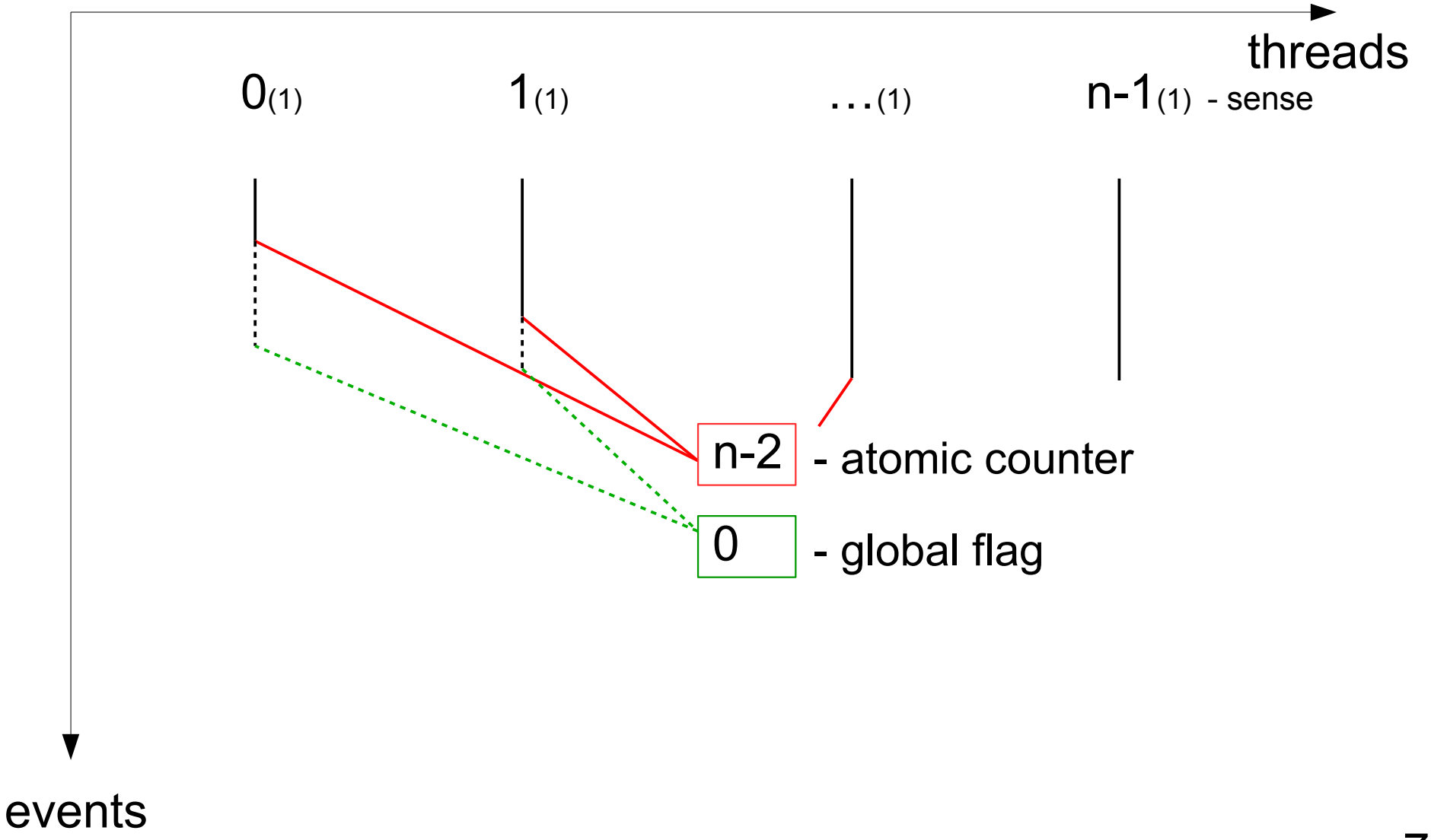
Centralized Barrier Algorithm



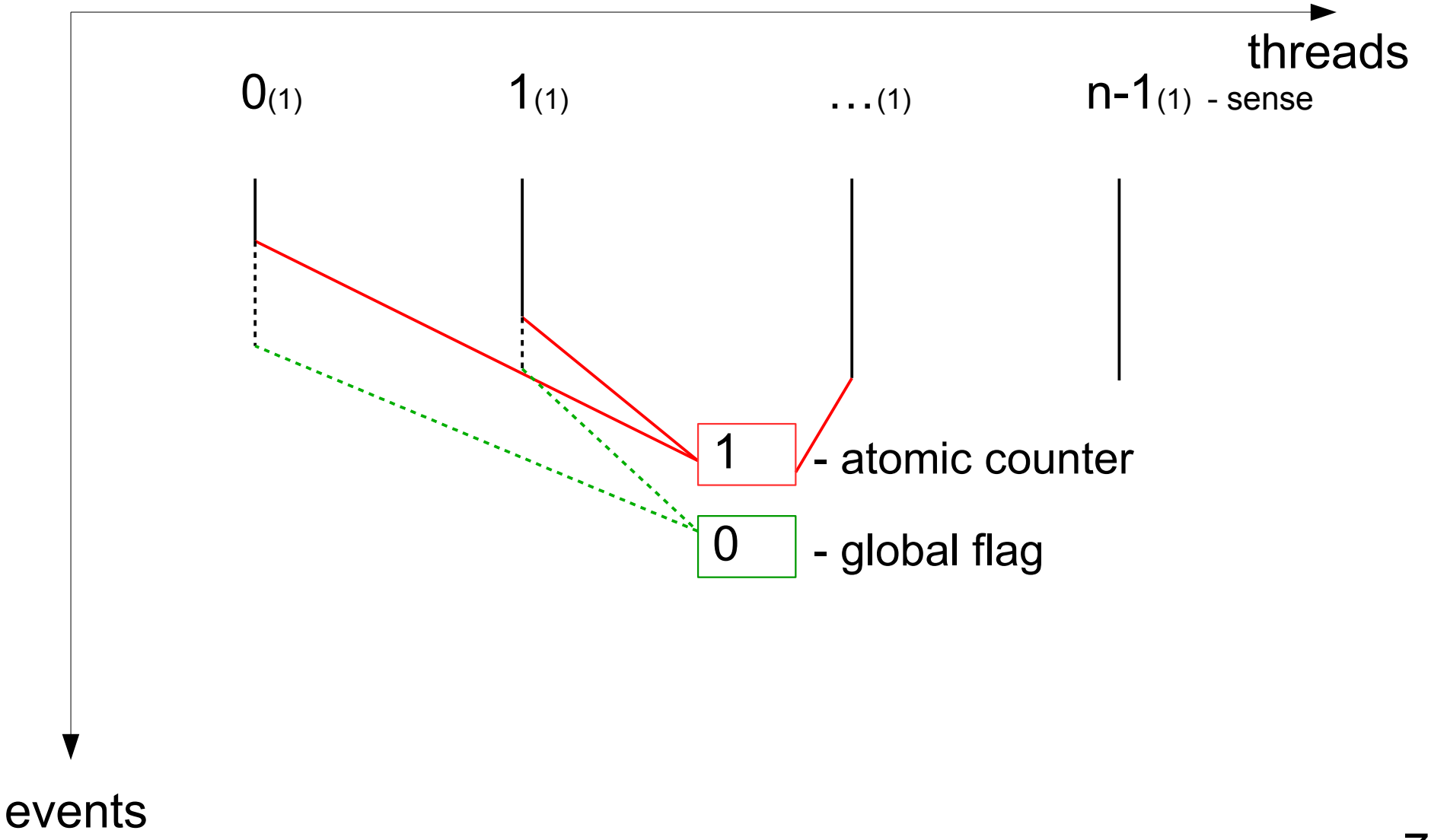
Centralized Barrier Algorithm



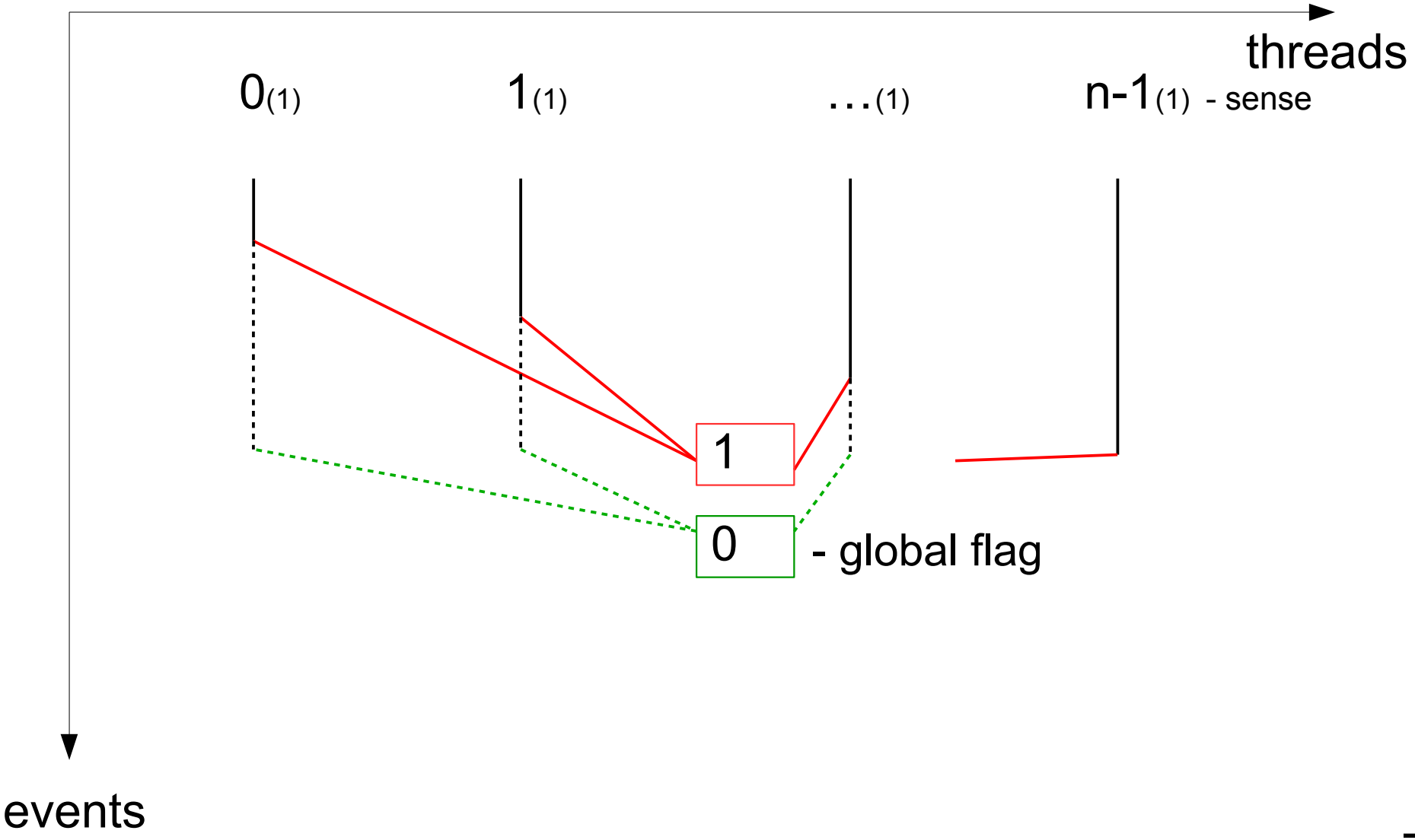
Centralized Barrier Algorithm



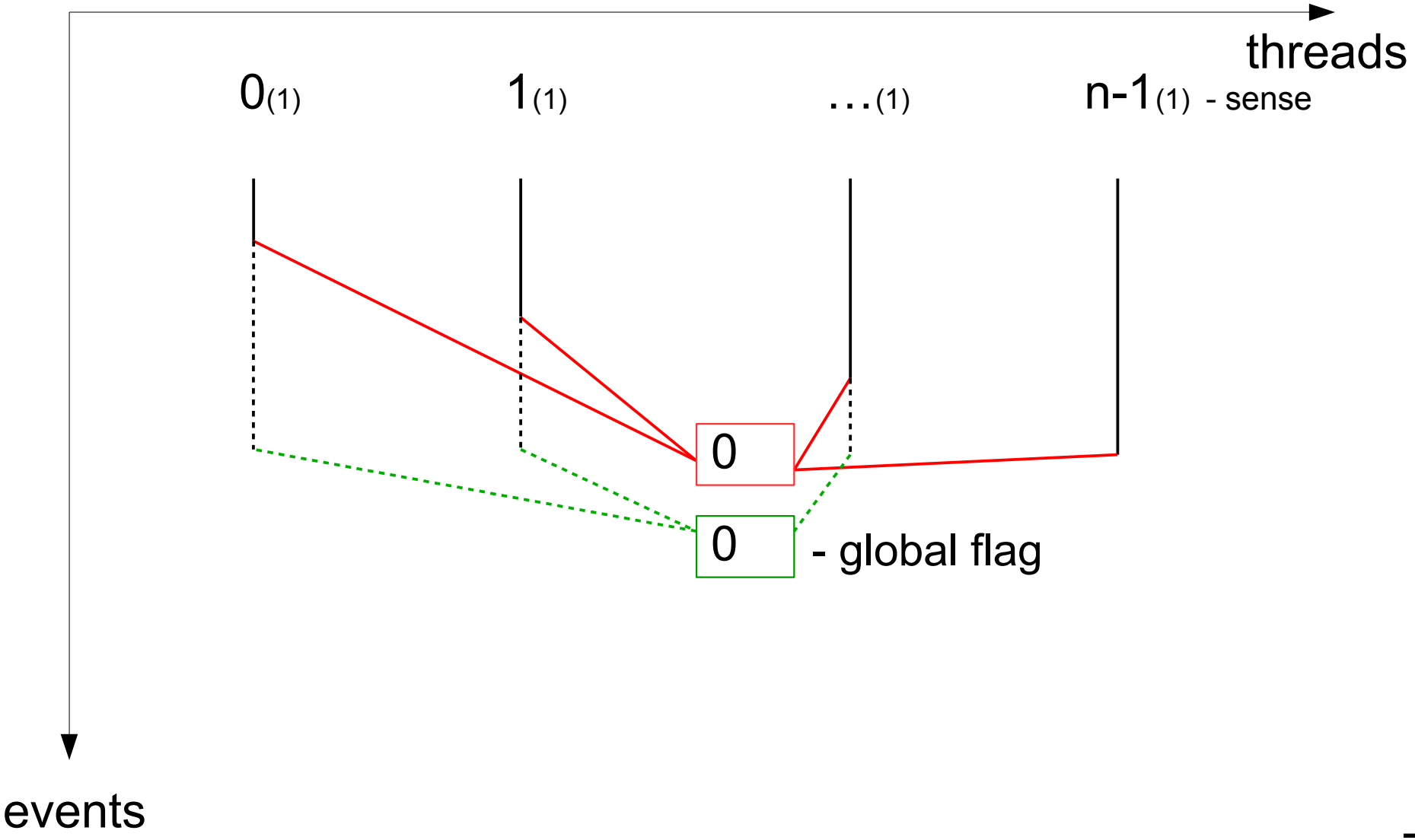
Centralized Barrier Algorithm



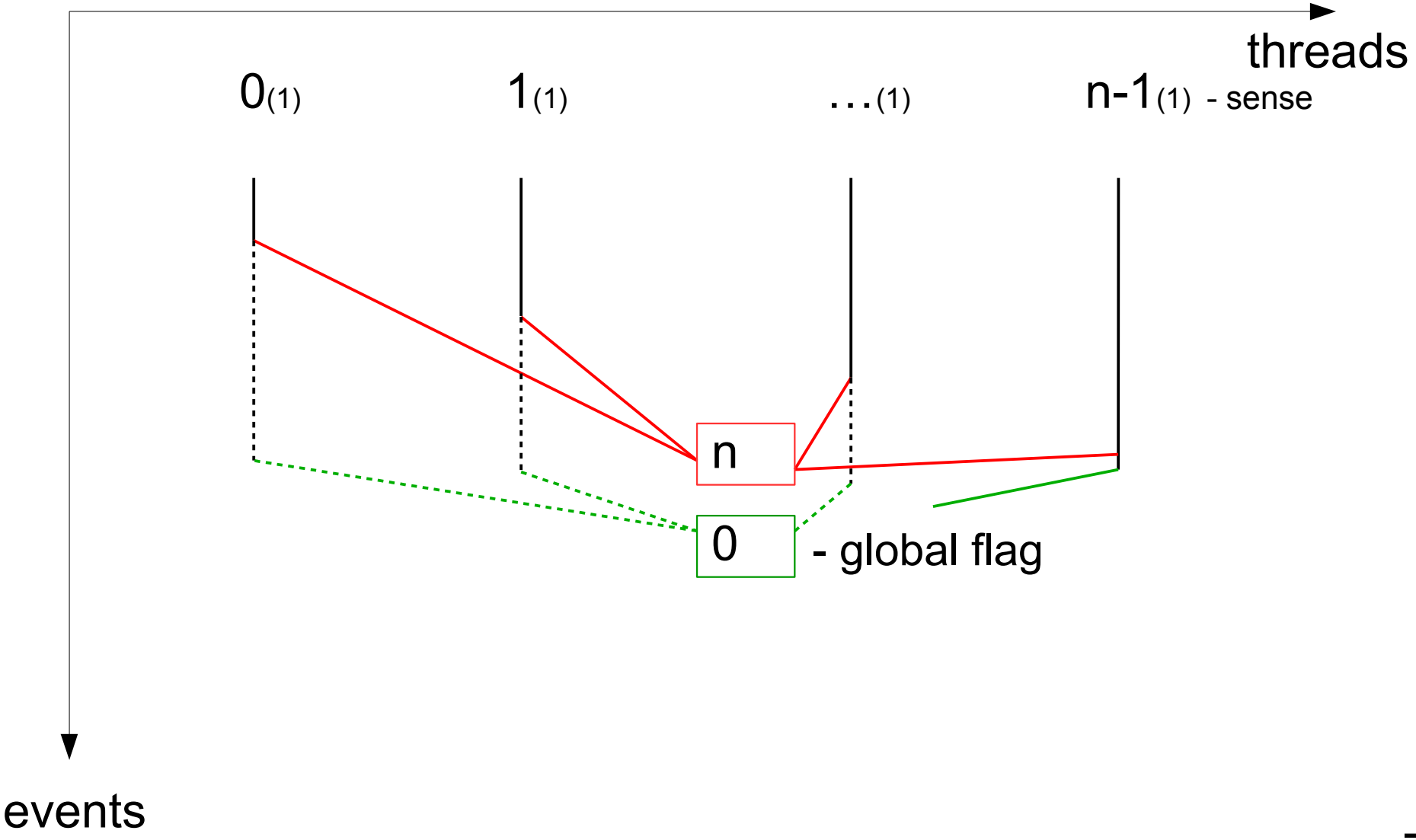
Centralized Barrier Algorithm



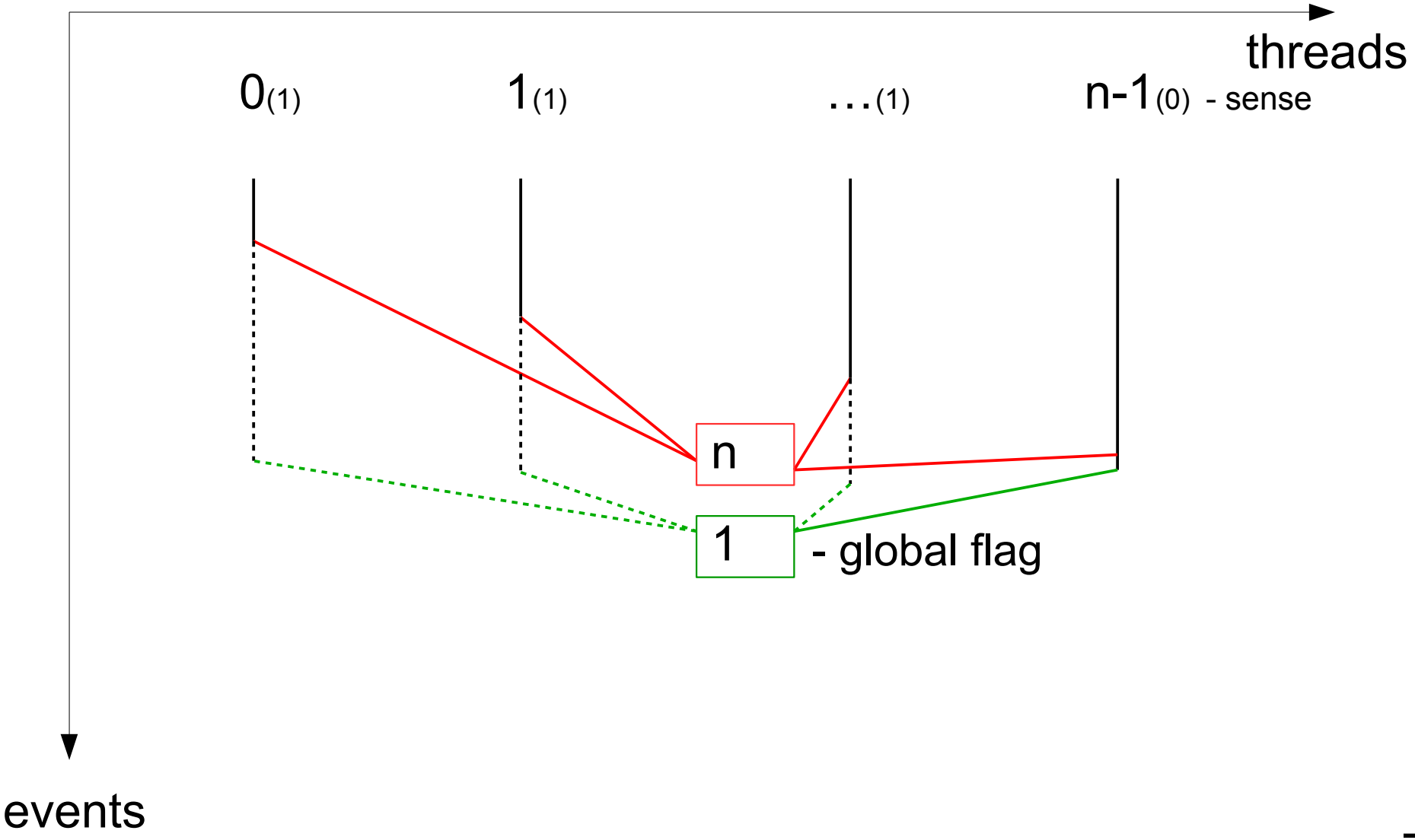
Centralized Barrier Algorithm



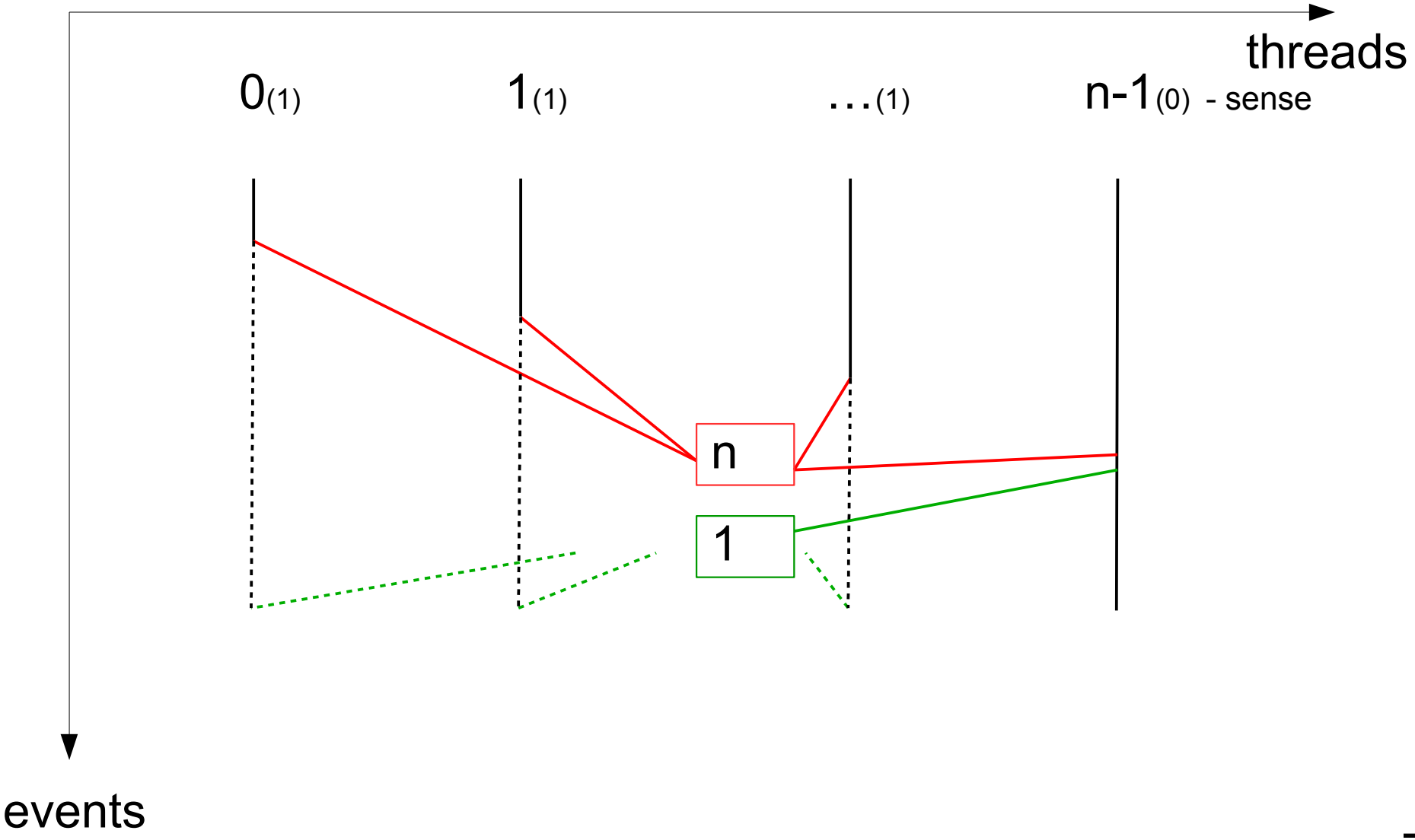
Centralized Barrier Algorithm



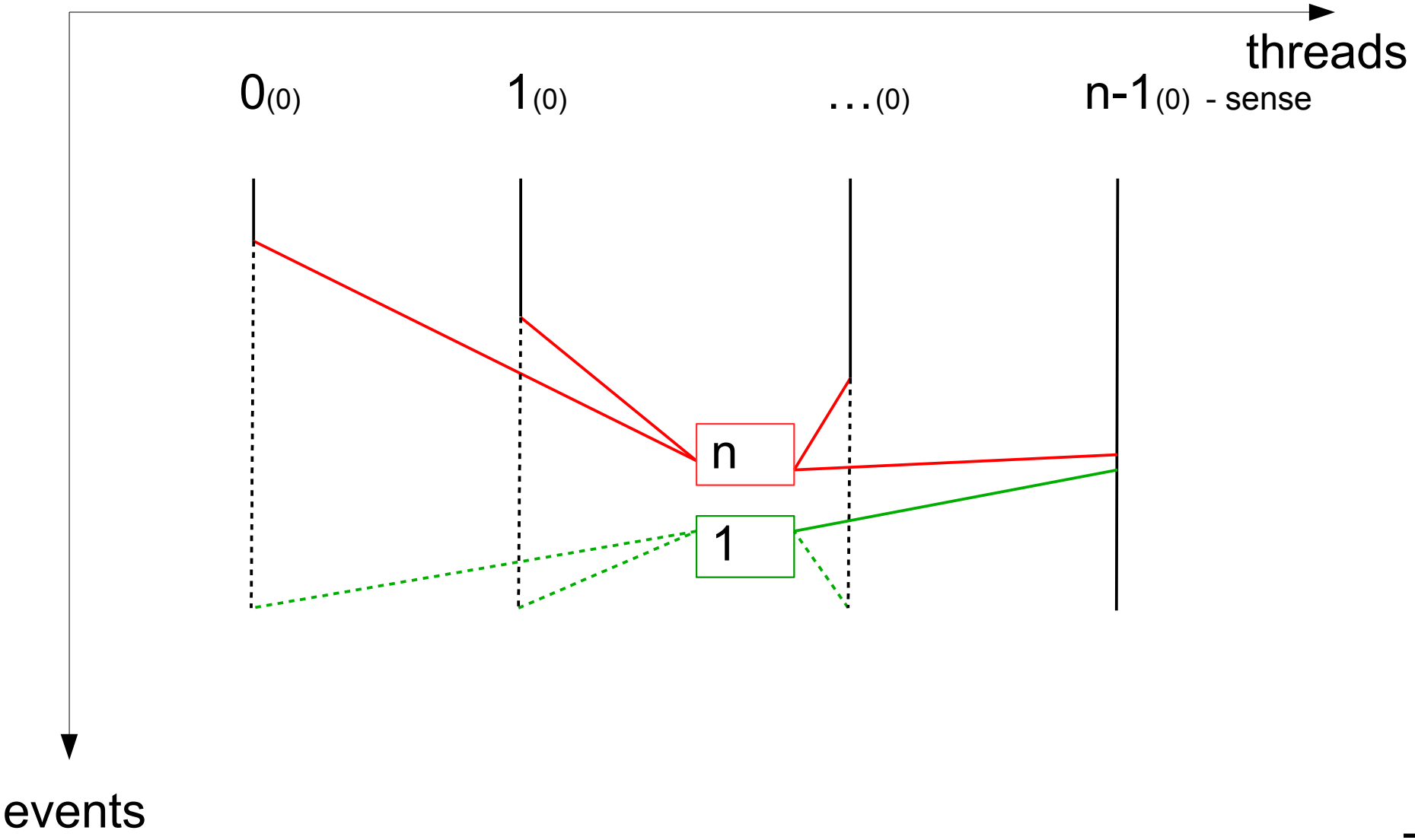
Centralized Barrier Algorithm



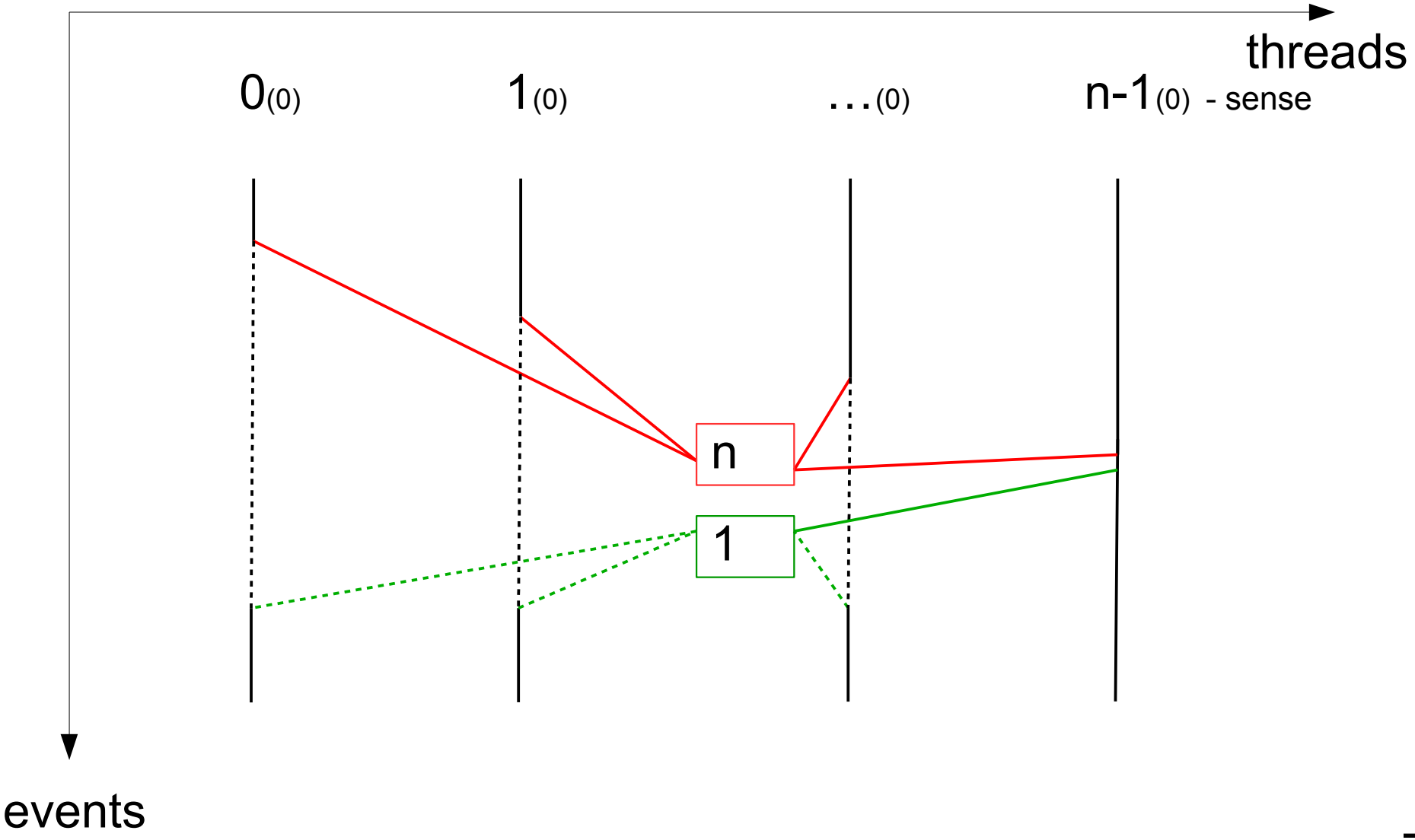
Centralized Barrier Algorithm



Centralized Barrier Algorithm

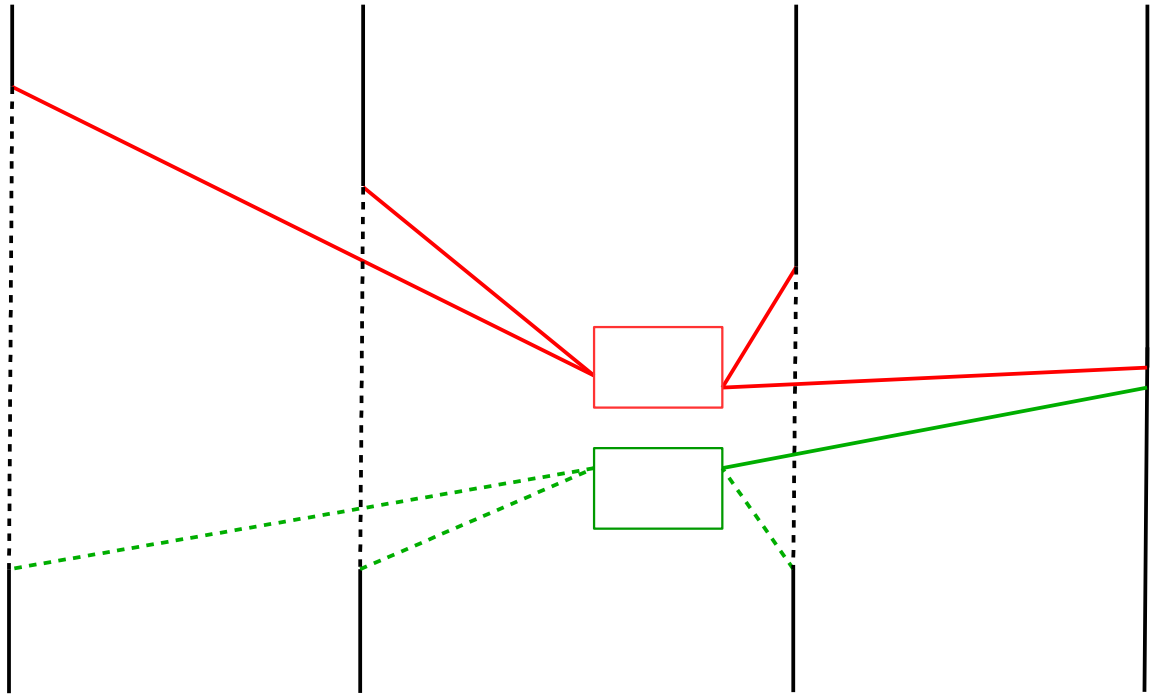


Centralized Barrier Algorithm



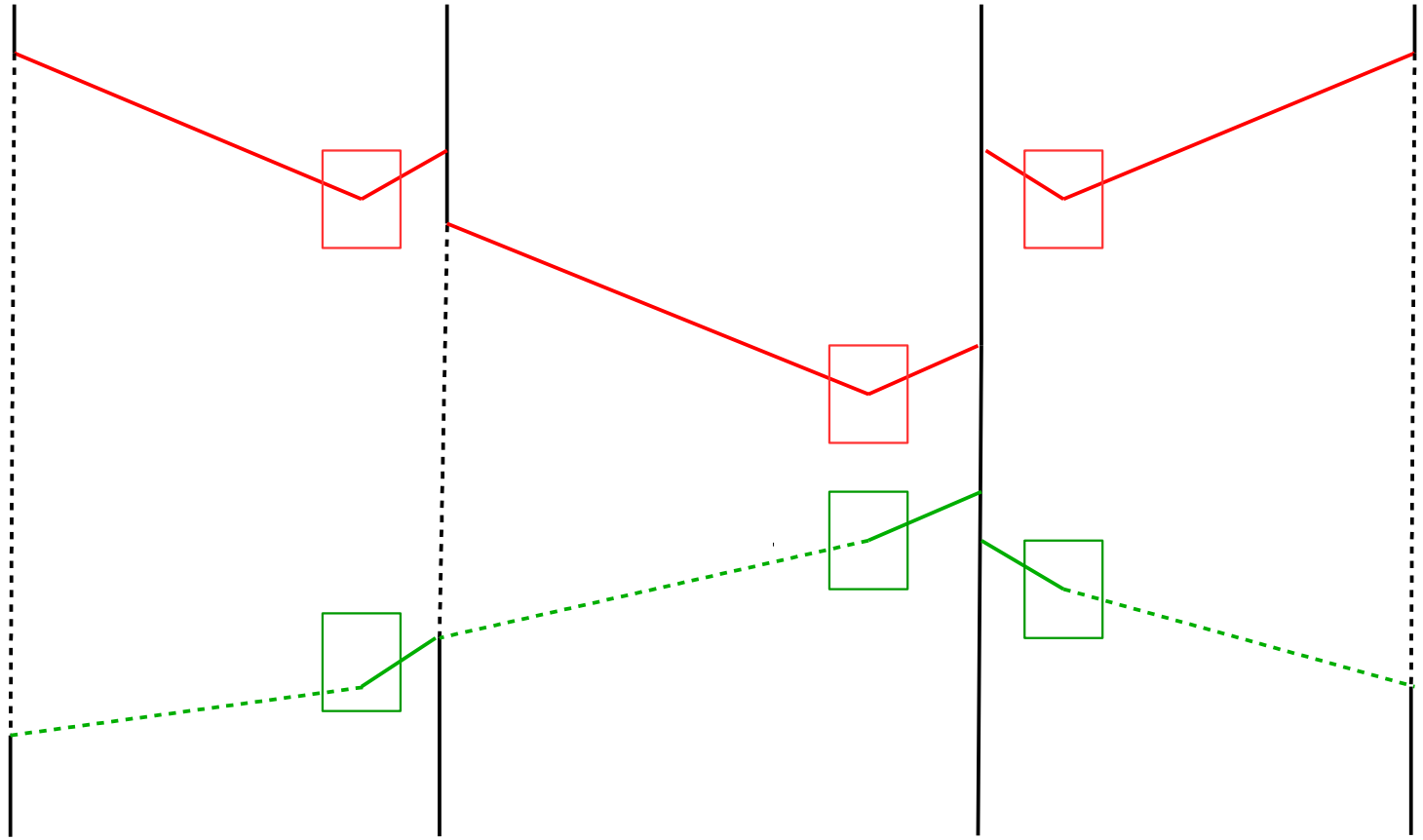
Centralized Barrier Algorithm

MANCHESTER 1824
The University of Manchester

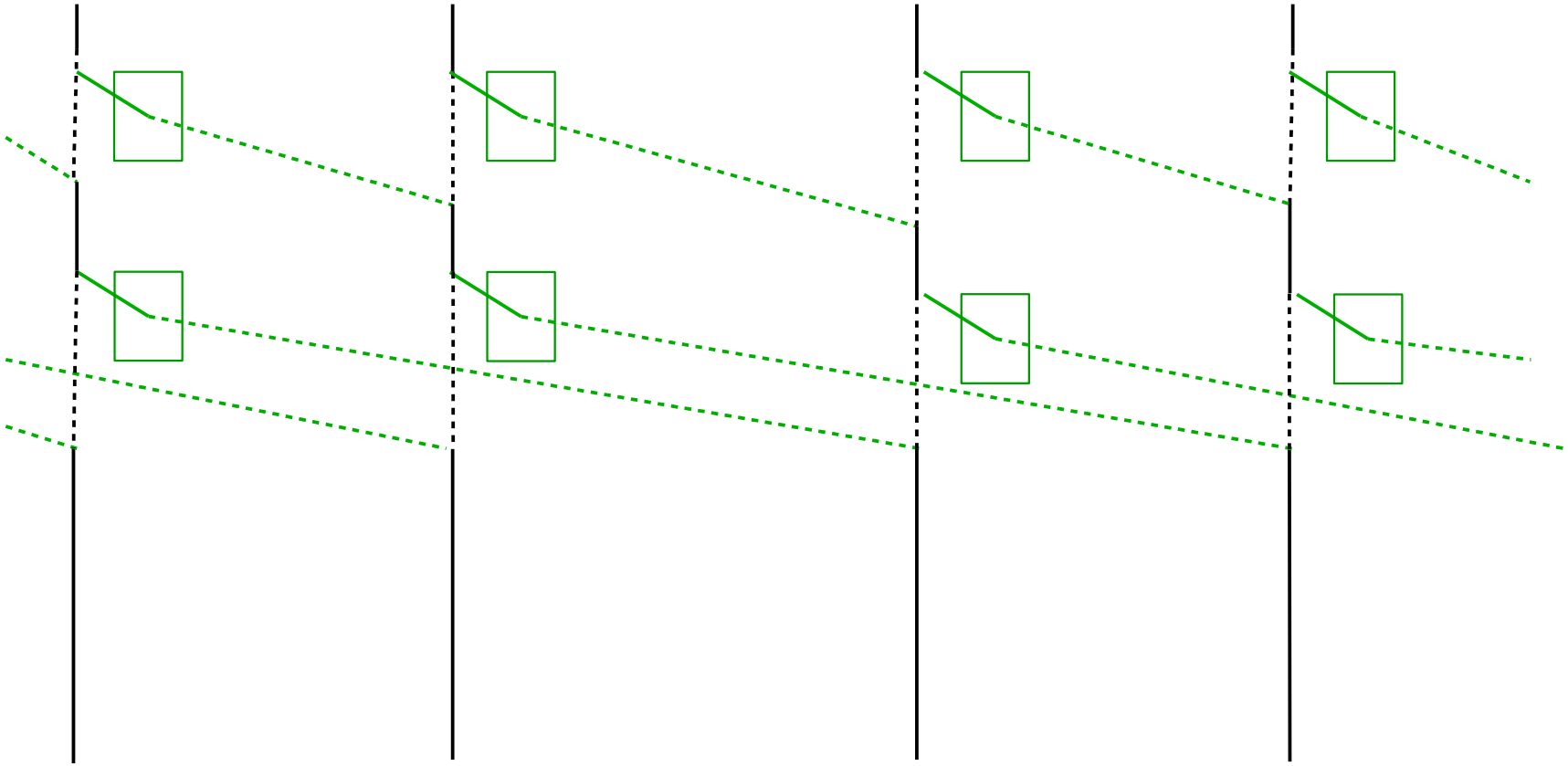


Combining Tree Barrier

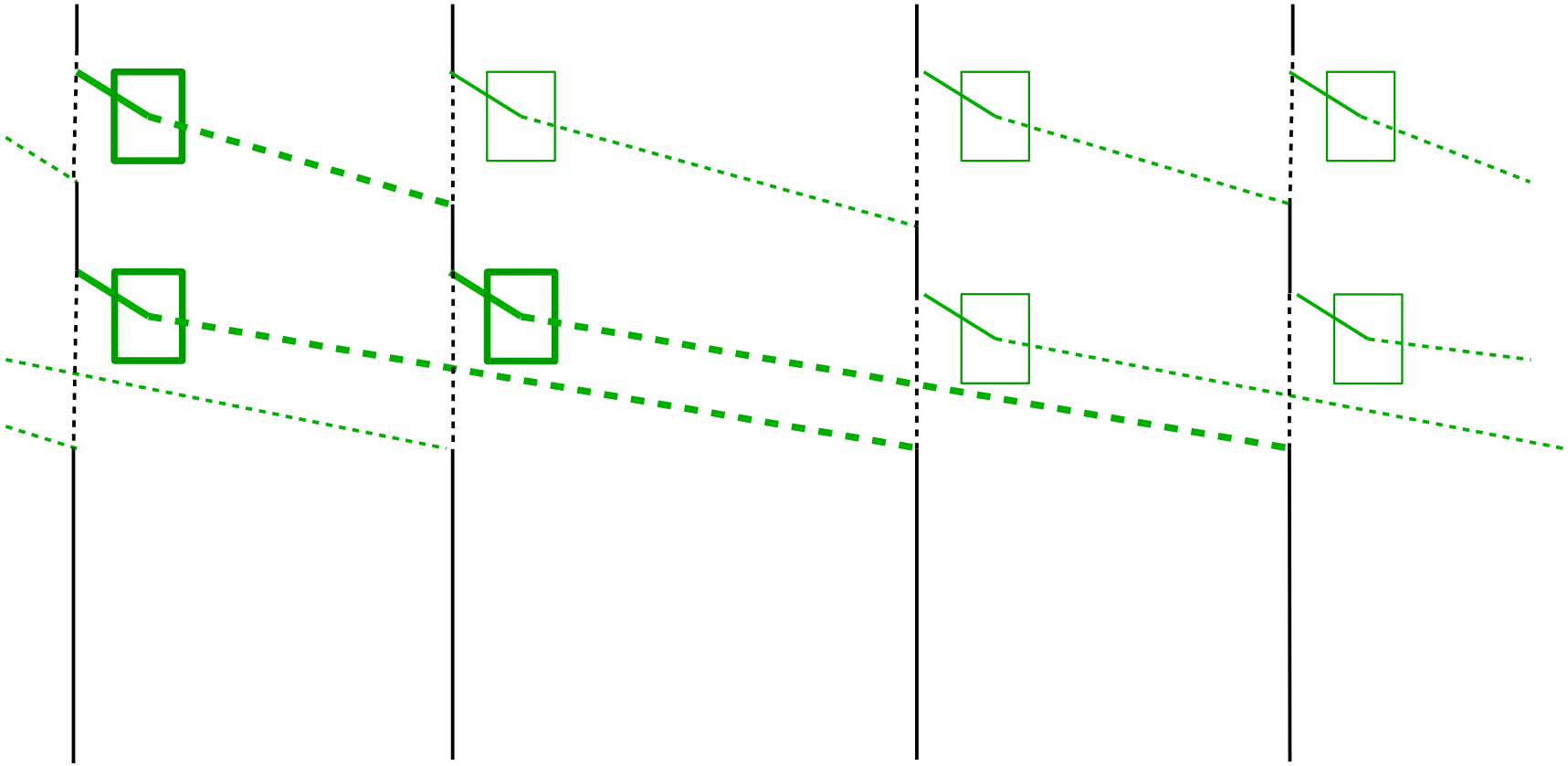
MANCHESTER 1824
The University of Manchester



Dissemination Barrier



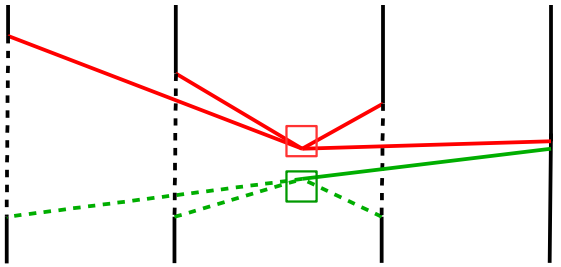
Dissemination Barrier



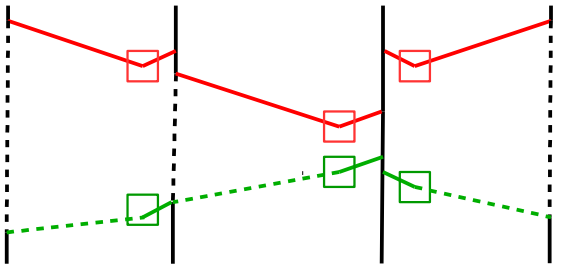
Barrier Synchronization Algorithms

MANCHESTER 1824
The University of Manchester

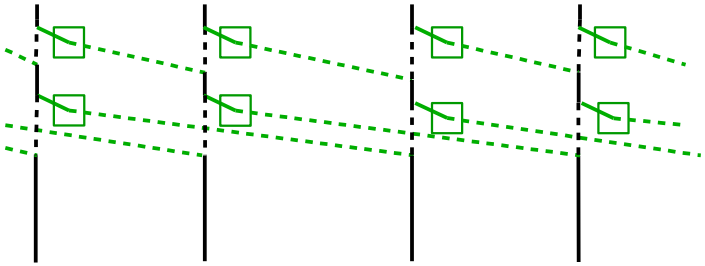
- Sense-reversing centralized barrier



- Combining tree barrier
- Static tournament
- Dynamic tournament



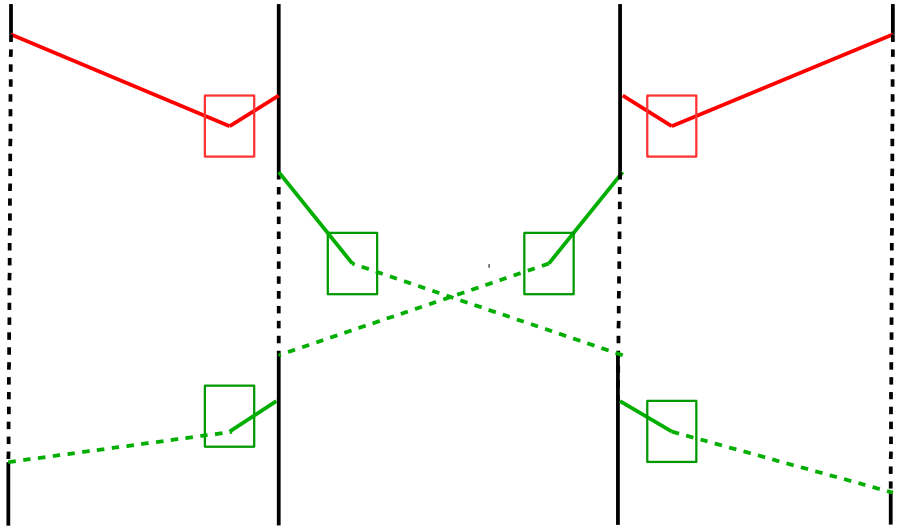
- Dissemination barrier



Hybrid Barrier Synchronization

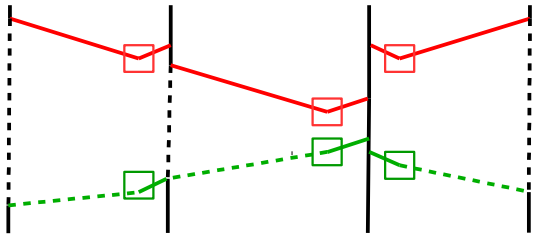
The University of Manchester

Hybrid Barrier



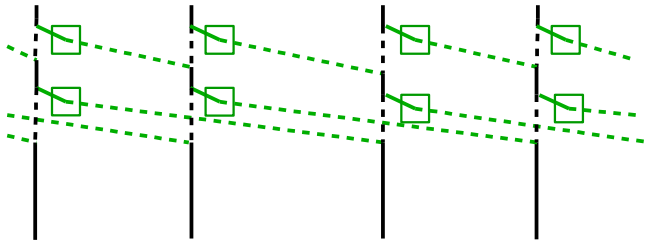
- Pros: hierarchical
- Neutral: r rounds, log(N) < r < 2 * log(N)

Tree Barrier



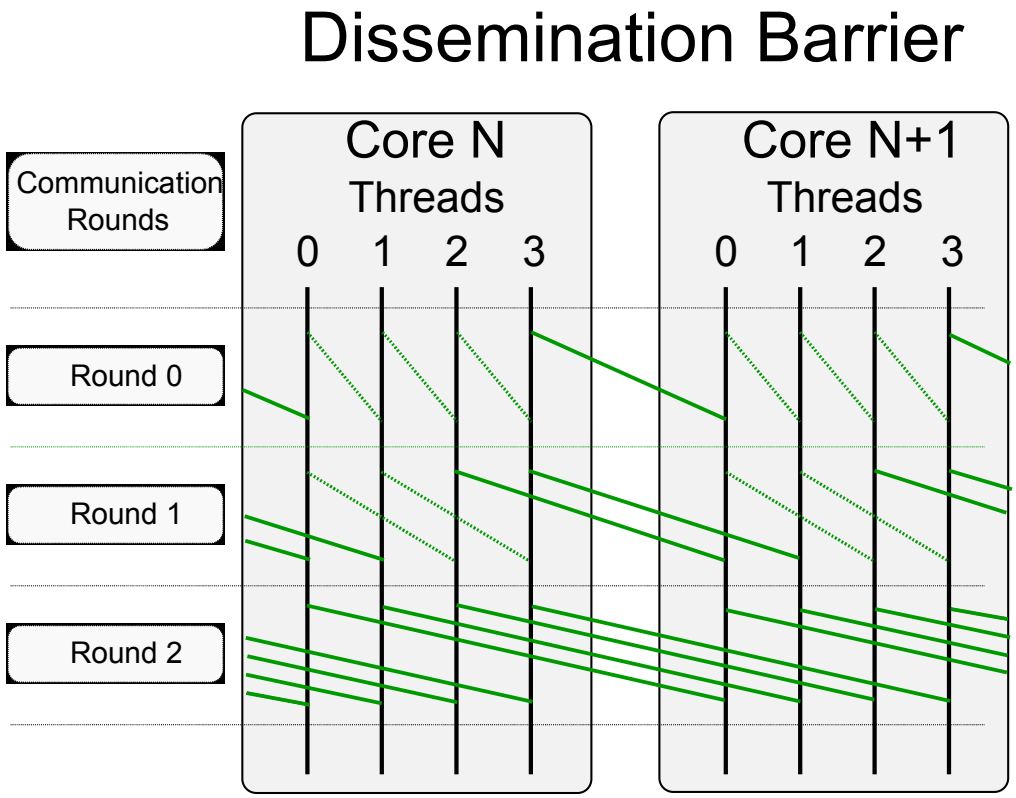
- Pros: hierarchical
- Cons: 2 * log(N) rounds

Dissemination Barrier

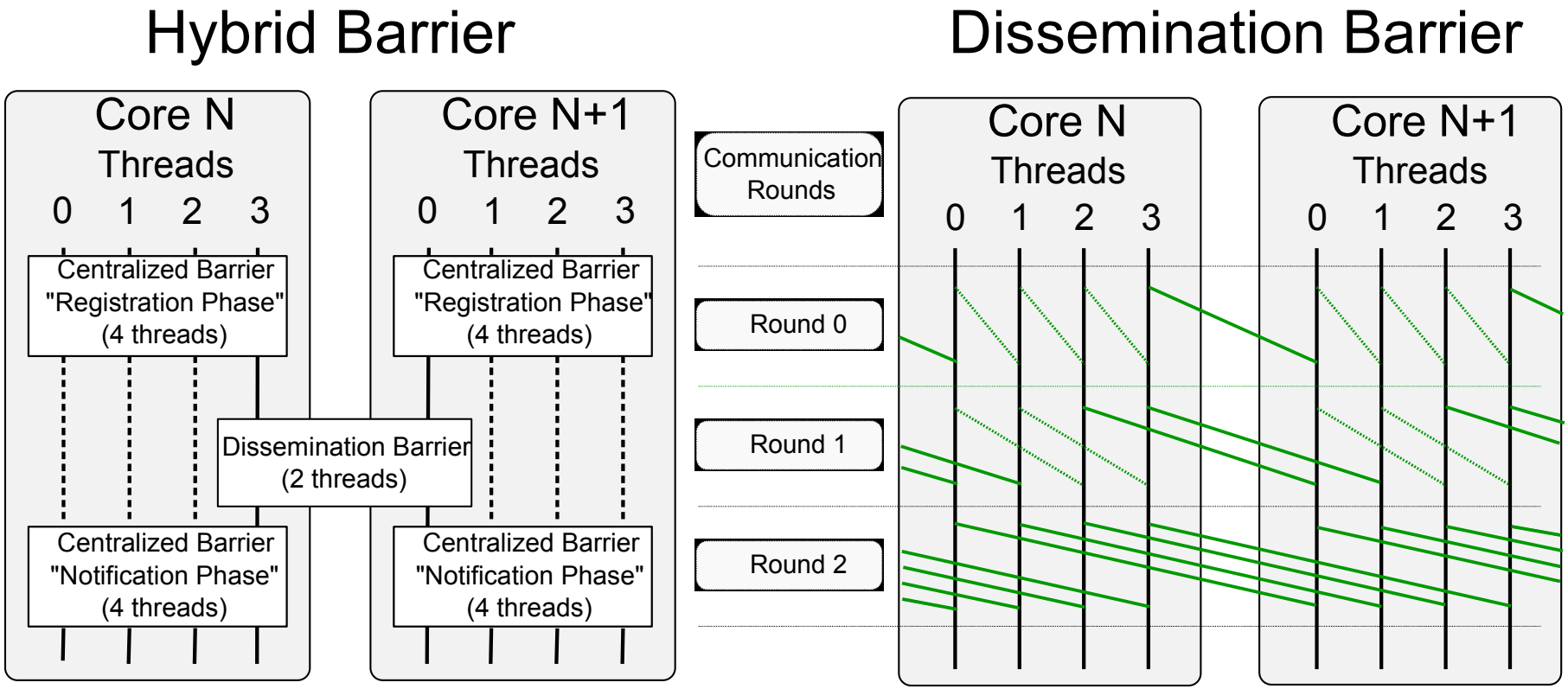


- Pros: log(N) rounds
- Cons: non-hierarchical

Rationale for Hybrid Barrier on Xeon Phi



Rationale for Hybrid Barrier on Xeon Phi



Benchmarks

- EPCC OpenMP Microbenchmarks
  - evaluating the overhead of the standalone barrier
- NAS Parallel Benchmarks
  - CG and MG
- Direct N-body Simulation
  - 1 particle per thread
  - highest possible barrier frequency

Experimental Set-Up

- Tested Configurations

  `<algorithm notification>_<delay>_<arity>`

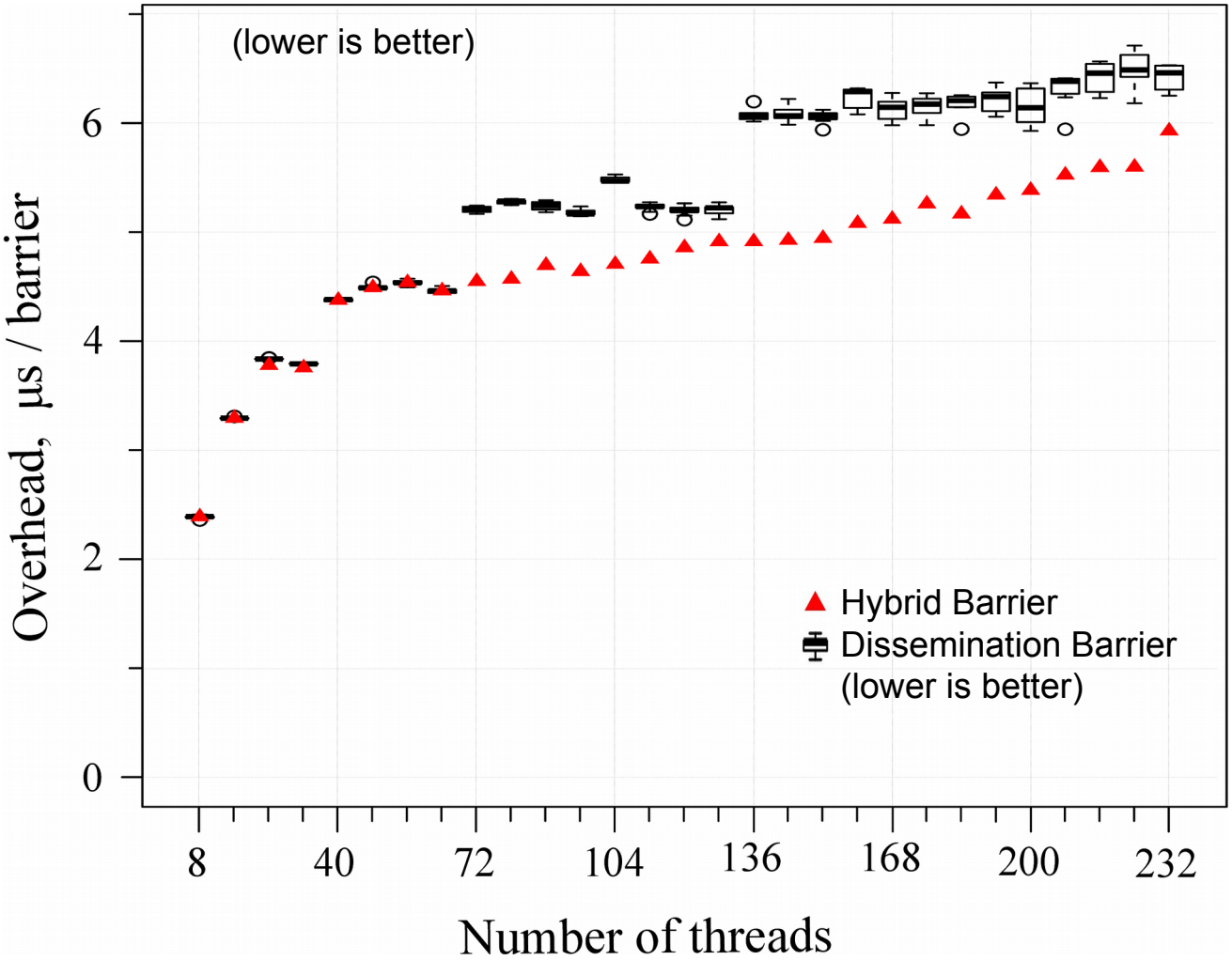
  algorithms: sr (centralized), dsmn (dissemination),
  dsmnH (hybrid), ct(combining tree),
  stn (static tournament), dtn(dynamic tournament),
  omp (intel OpenMP barrier)
  notification: ls (tree-based local flags), gs (global flag)
  delay: spin, pause (64-cycle delay)
  arity: 2, 3, 4, 8, 16, 32 for ct (combining tree)
  2, 3, 4, 5 for stn (static tournament)
  2, 3, 4 for dtn (dynamic tournament)

- Number of Threads
  - 8 to 232 with a step 8

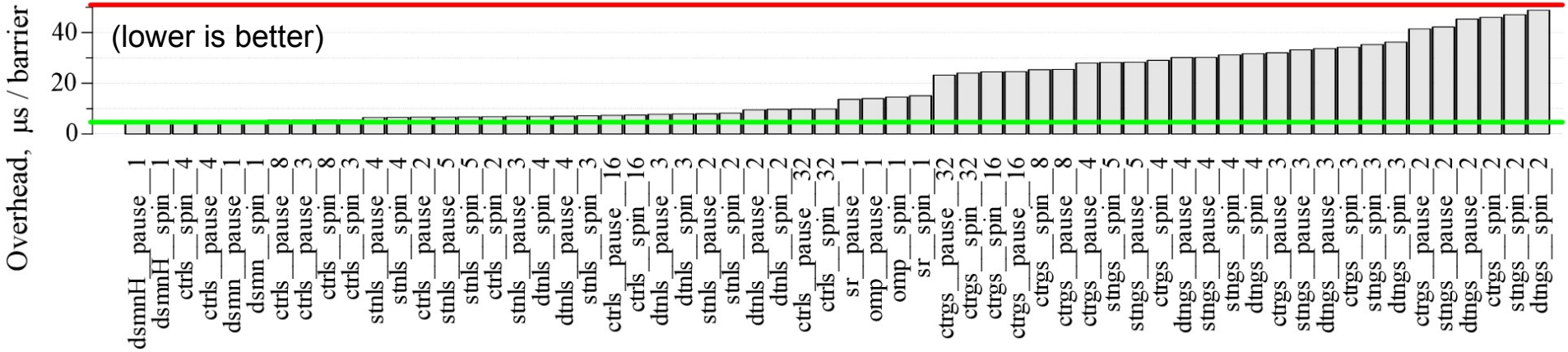
- Evaluation Metric
  - geomean across the number of threads

Dissemination Barrier vs Hybrid Barrier

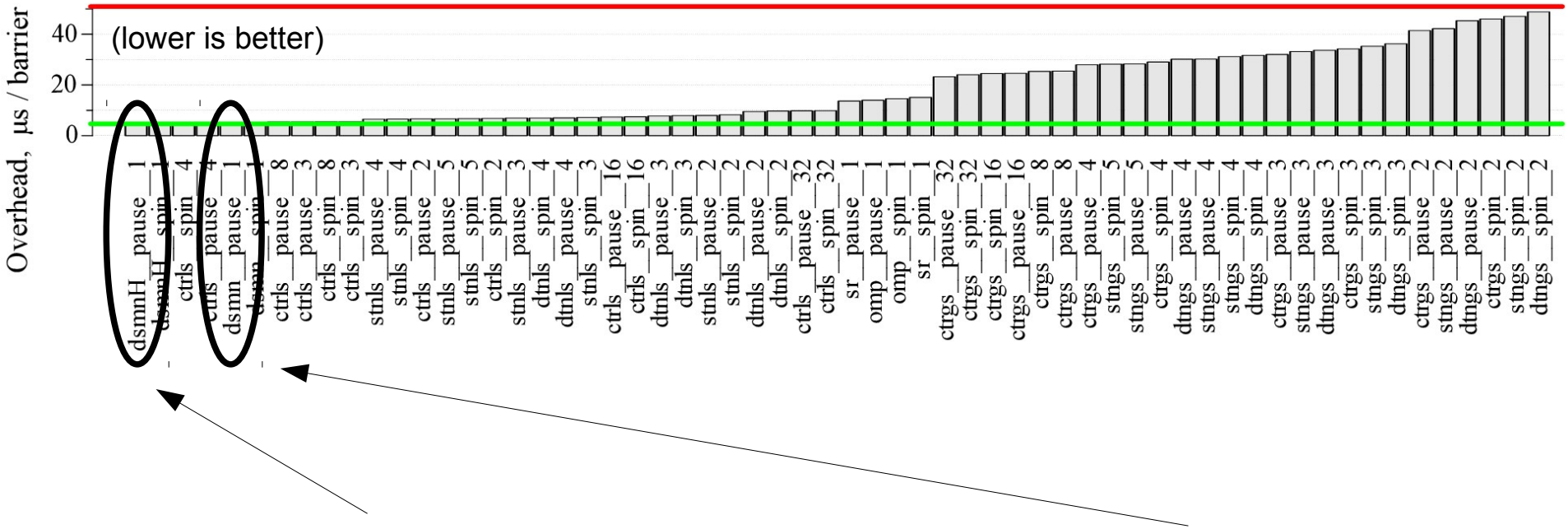
EPCC Barrier Microbenchmark



Geomean Overhead on EPCC

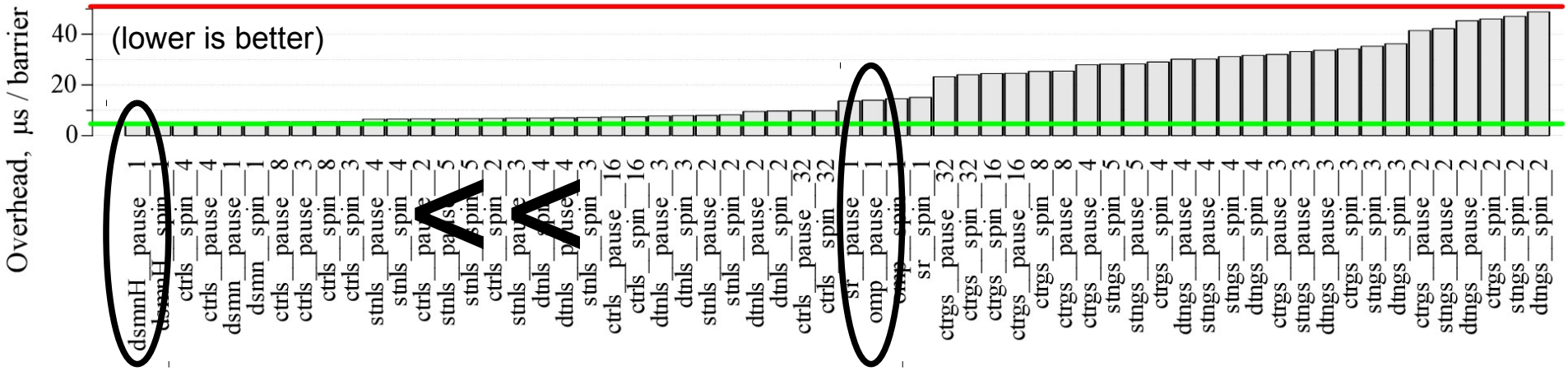


Geomean Overhead on EPCC

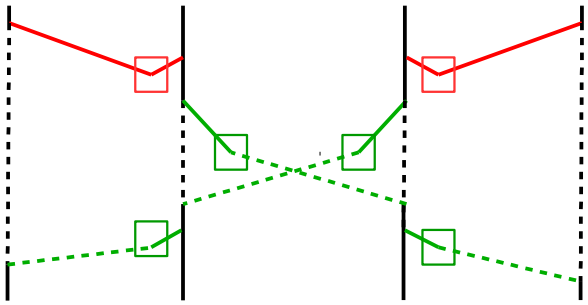


hybrid barrier < dissemination barrier

Geomean Overhead on EPCC

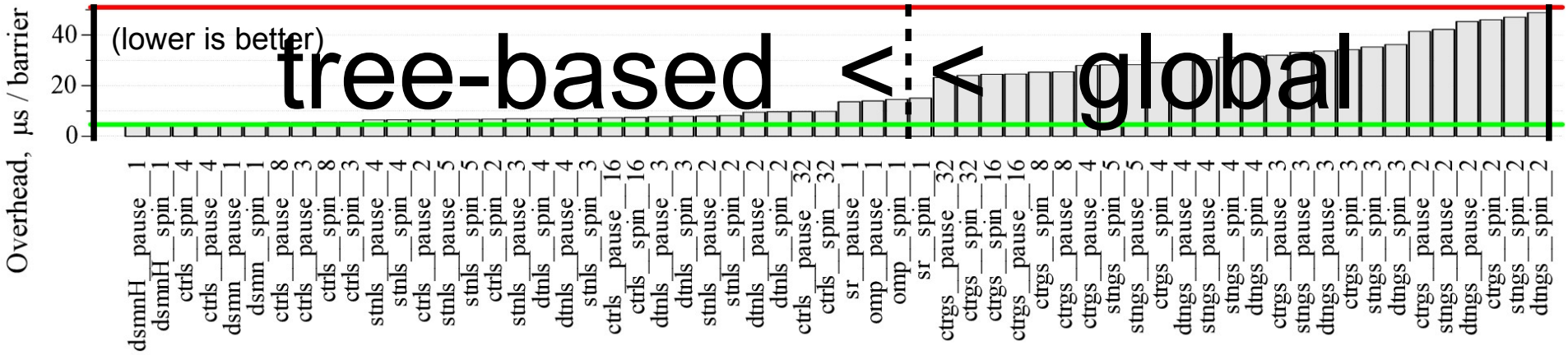


Key Observations: **hybrid barrier is 3x better than omp barrier**

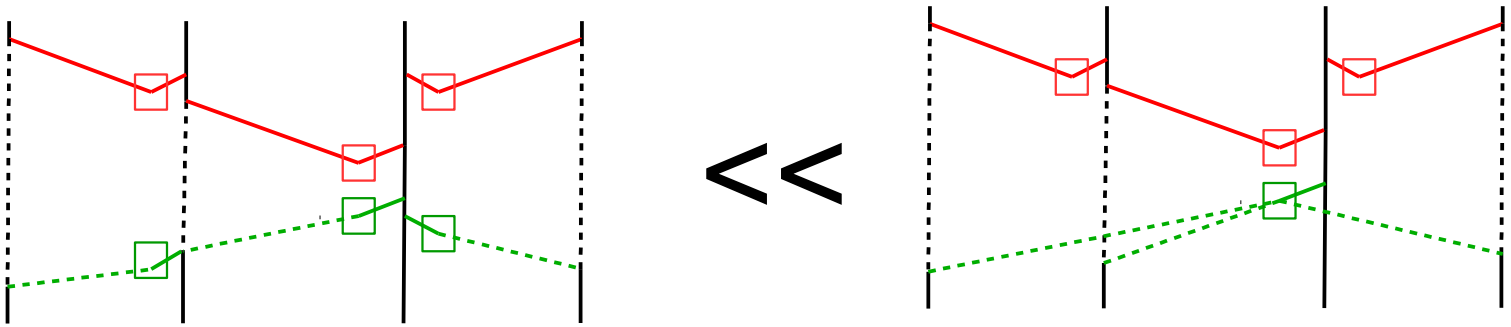


<< omp barrier

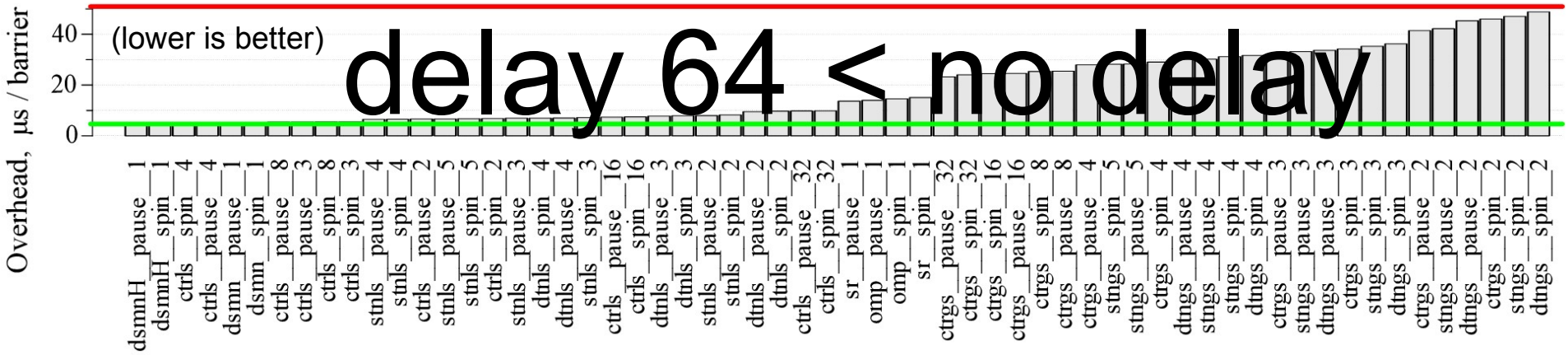
Geomean Overhead on EPCC



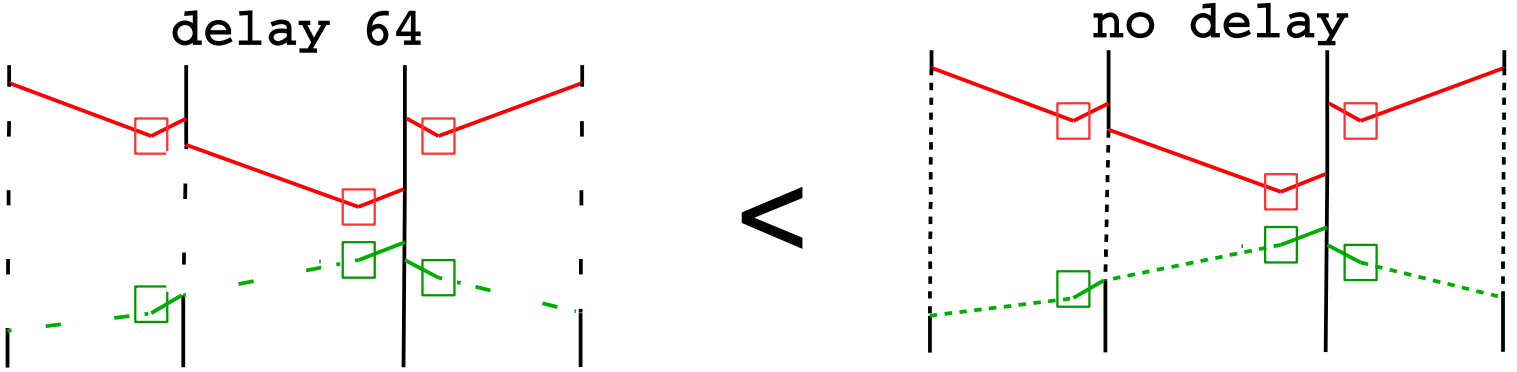
Key Observations: **barriers with tree-based notification have lower overhead than with global flag notification**



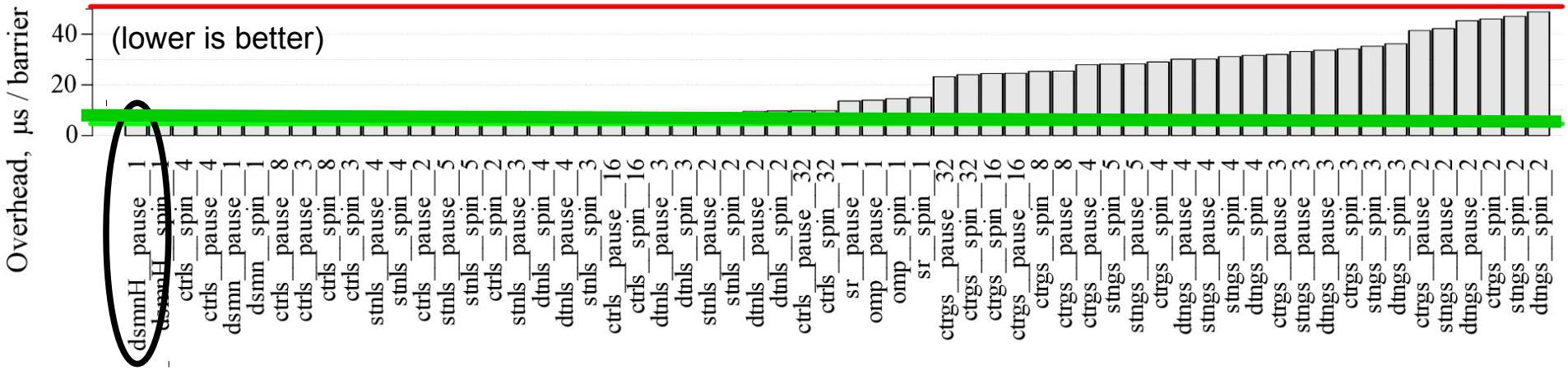
Geomean Overhead on EPCC



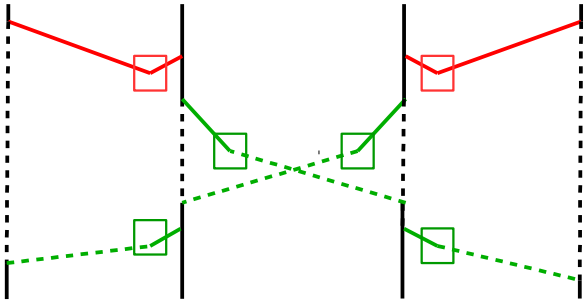
Key Observations: **barrier with delayed busy-waiting outperforms the same barrier with non-delayed spinning**



Geomean Overhead on EPCC

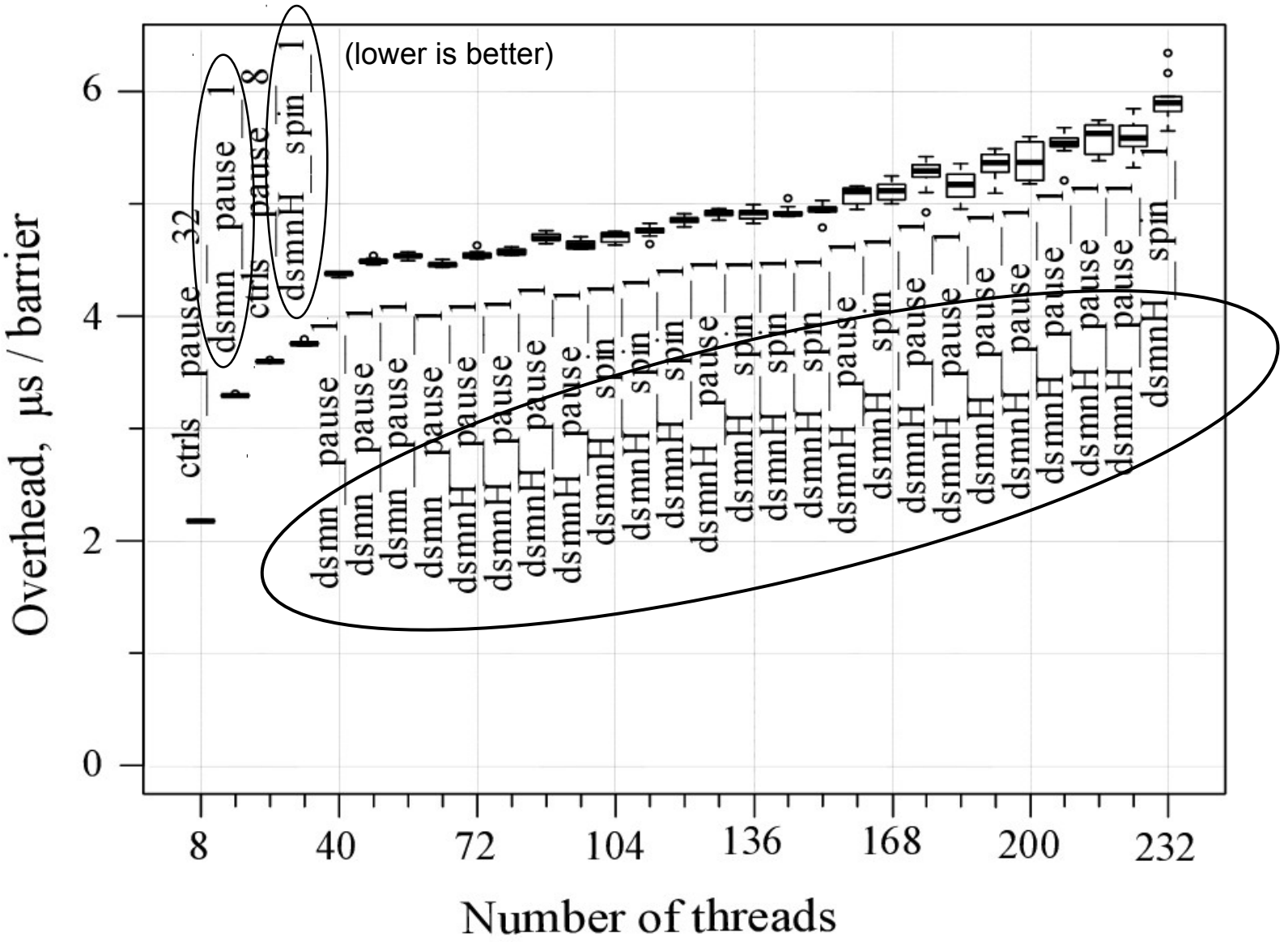


Key Observations: **a statically chosen hybrid algorithm is only marginally outperformed by an algorithm versioned for a given number of threads**



~ best_barrier(N_thr)

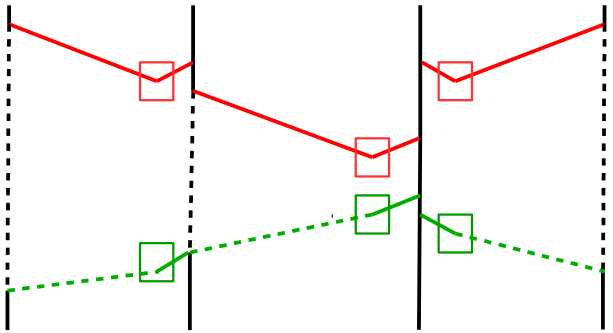
Best Algorithm on EPCC for N Threads



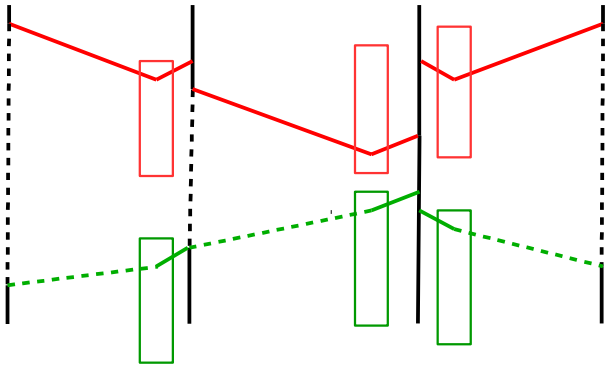
Effect of Specific Store Instructions

MANCHESTER 1824
The University of Manchester

- ordinary loads and stores (baseline)

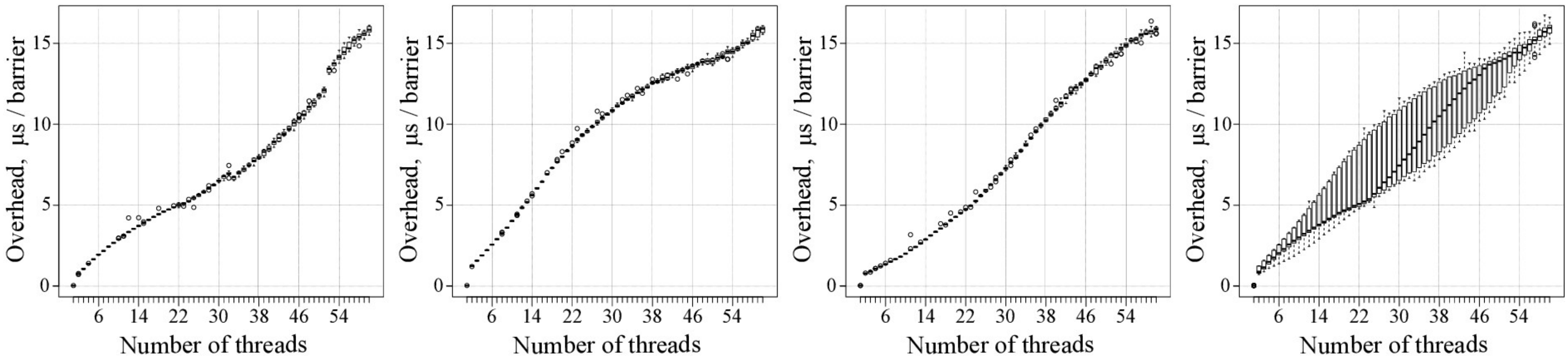


- non-globally ordered streaming stores: (same as baseline)
- globally ordered streaming stores: (~5% improvement)

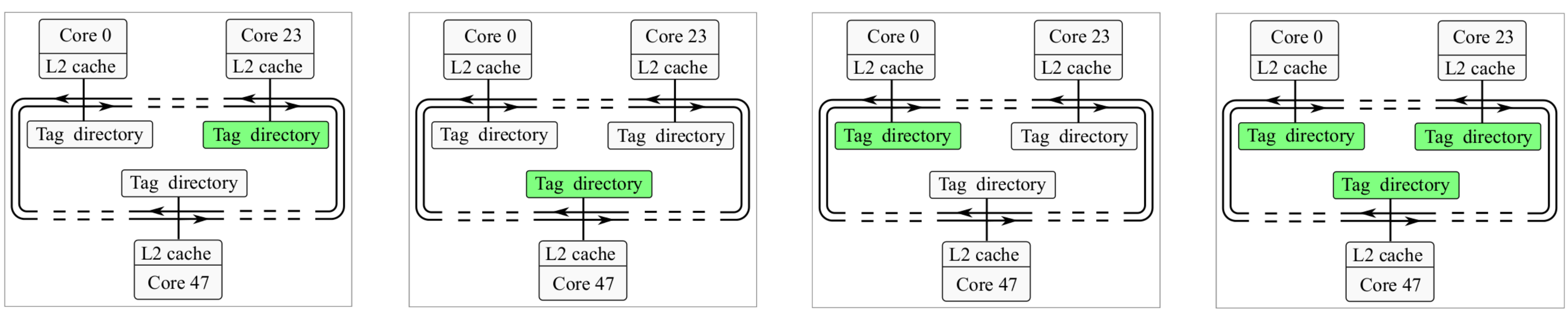


Effect of the Ring Interconnect

Impact of Non-uniform Cache Line Access Latency



(a) experiment 1 (b) experiment 2 (c) experiment 3 (d) experiments 1-3



Related Work

- SIMD barrier (Caballero *et al.,* 2013)
  - up to 2.84x lower overhead than Intel OpenMP barrier

- Barrier with NUCA-aware address selection (Dolbeau, 2014)
  - 2.85x lower overhead than the Intel OpenMP barrier with selection of addresses for inter-core communication variables

- Model for dissemination barrier (Ramos *et al.*, 2013)

Conclusions

- Thorough evaluation of barrier synchronization algorithms on Intel Xeon Phi
- Novel hybrid barrier algorithm proposed
  - 3x lower overhead than the Intel OpenMP barrier
- Analysis of key specificities of Intel Xeon Phi
  - impact of the ring interconnect
  - impact of delay instructions
  - impact of streaming stores
- Open-source evaluation framework is available at
  - `https://github.com/arodchen/cbarriers`