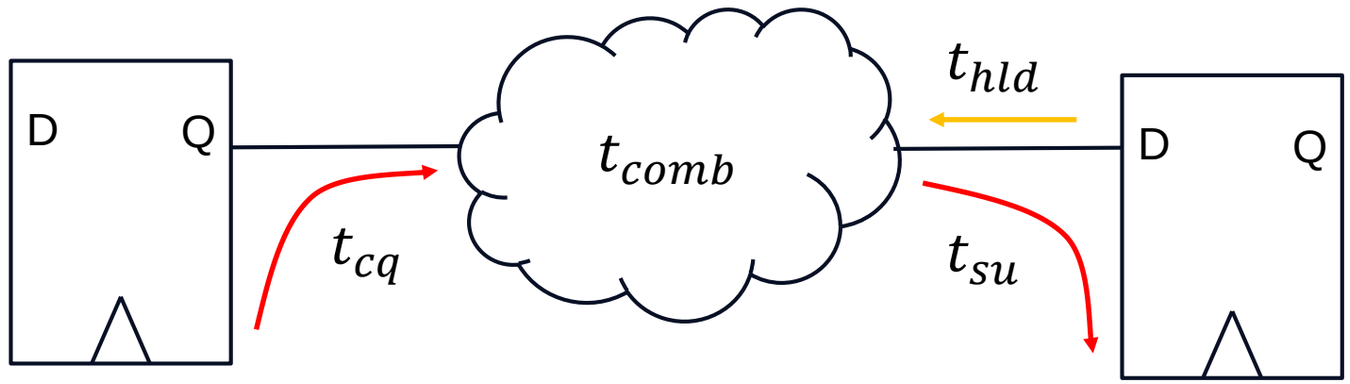
Tatum: Parallel Timing Analysis for Faster Design Cycles and Improved Optimization

Kevin E. Murray and Vaughn Betz

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

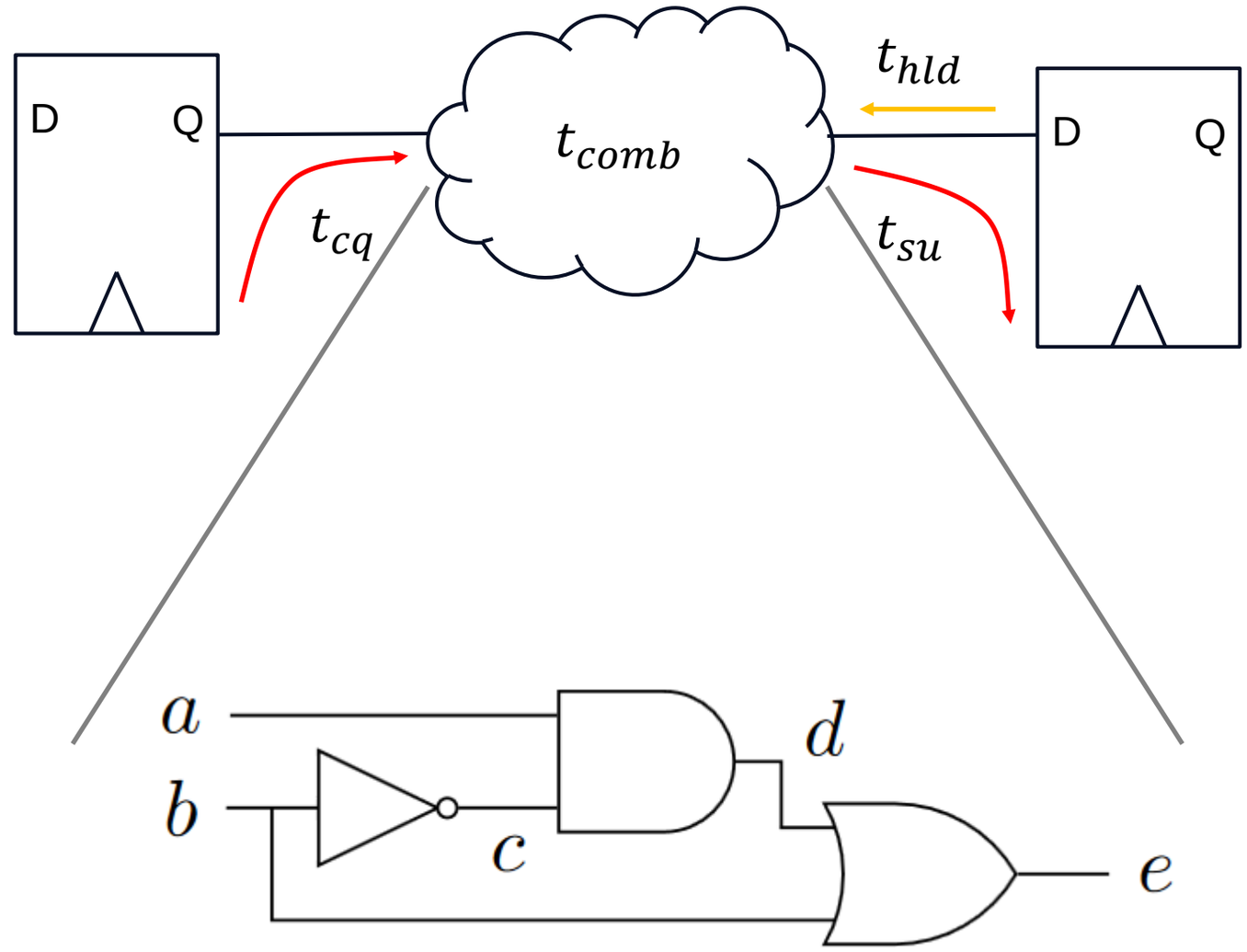
Timing Analysis



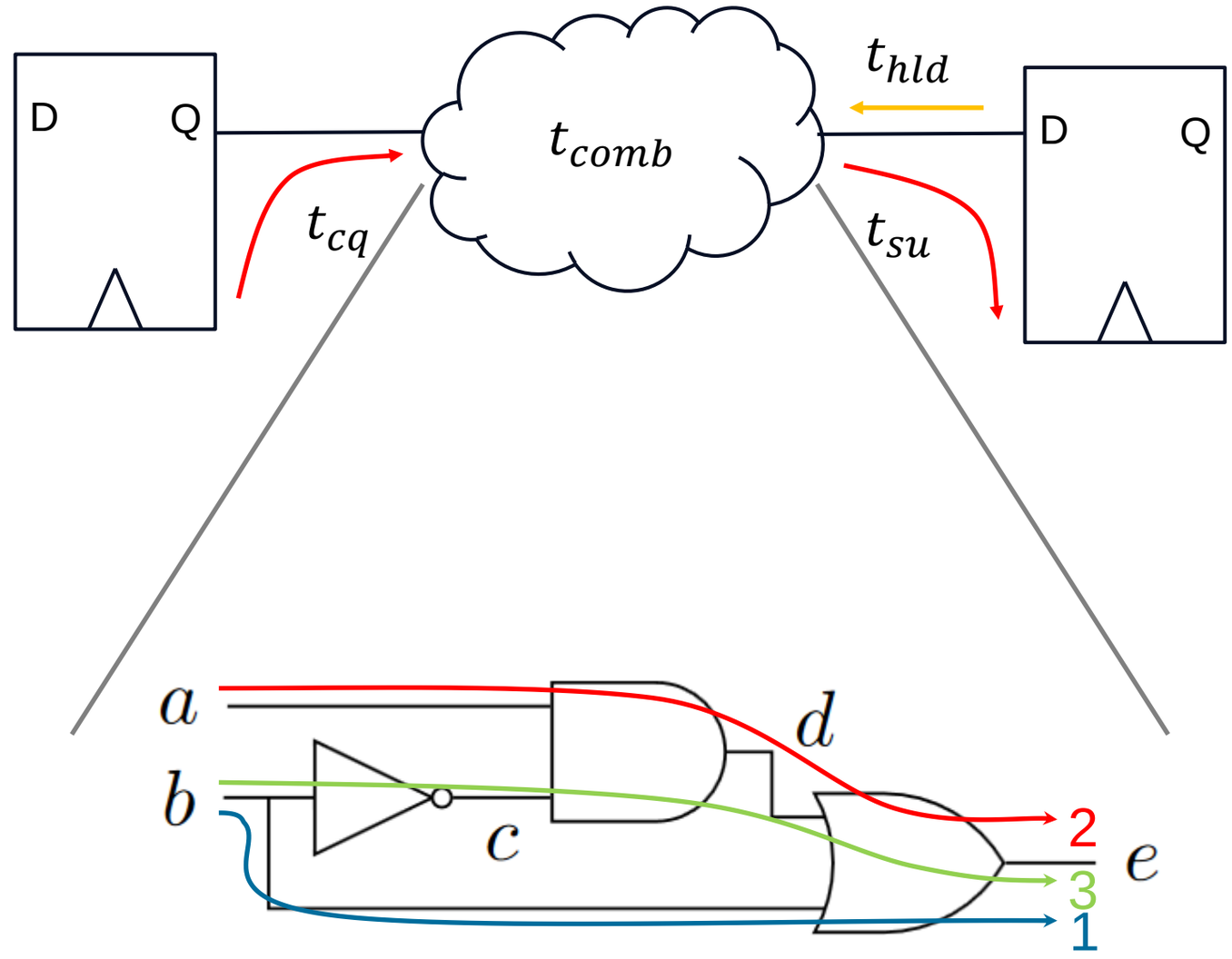
$$t_{cq} + t_{comb} + t_{su} \leq \frac{1}{f_{max}}$$

$$t_{cq} + t_{comb} \geq t_{hld}$$

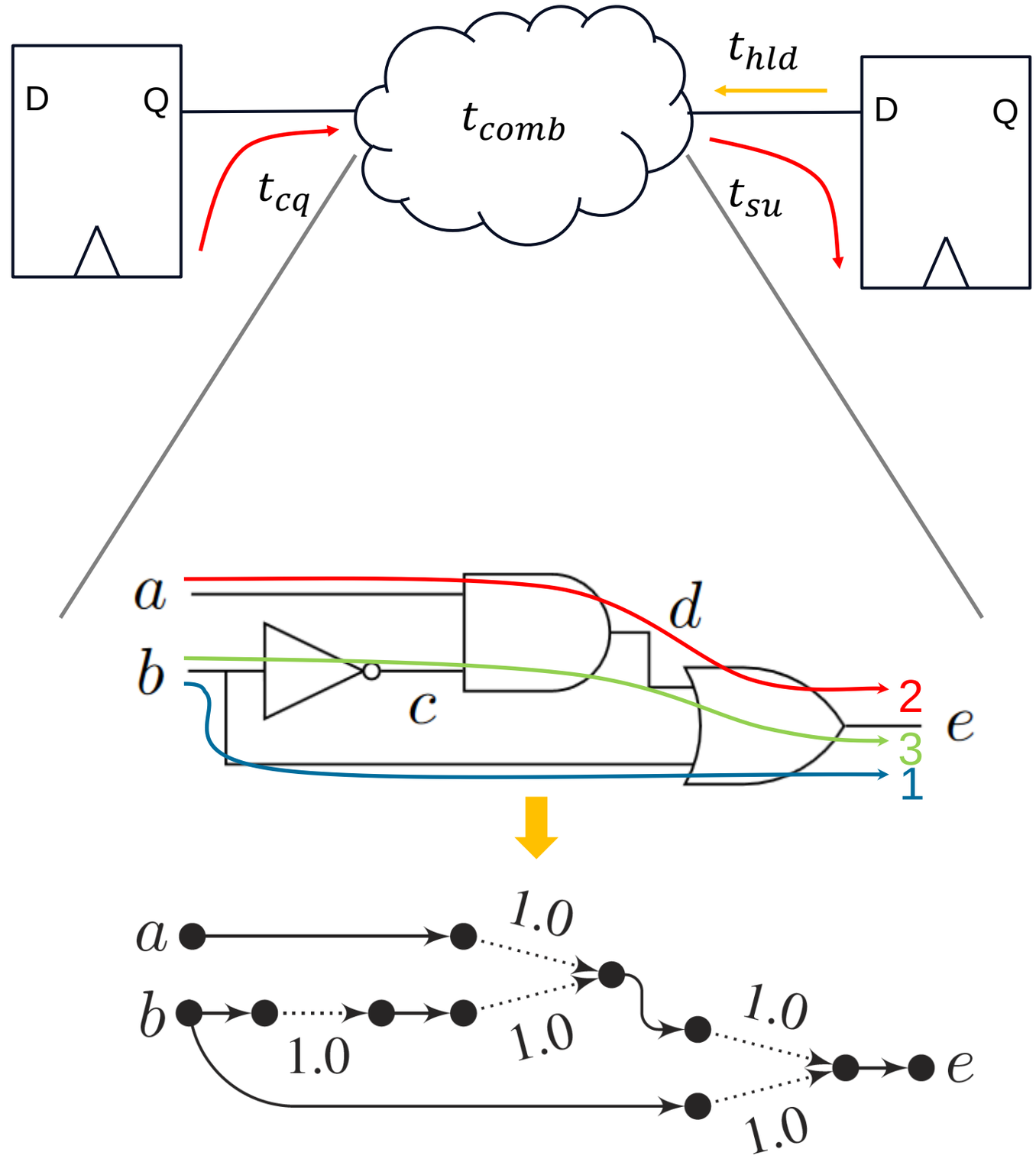
Timing Analysis



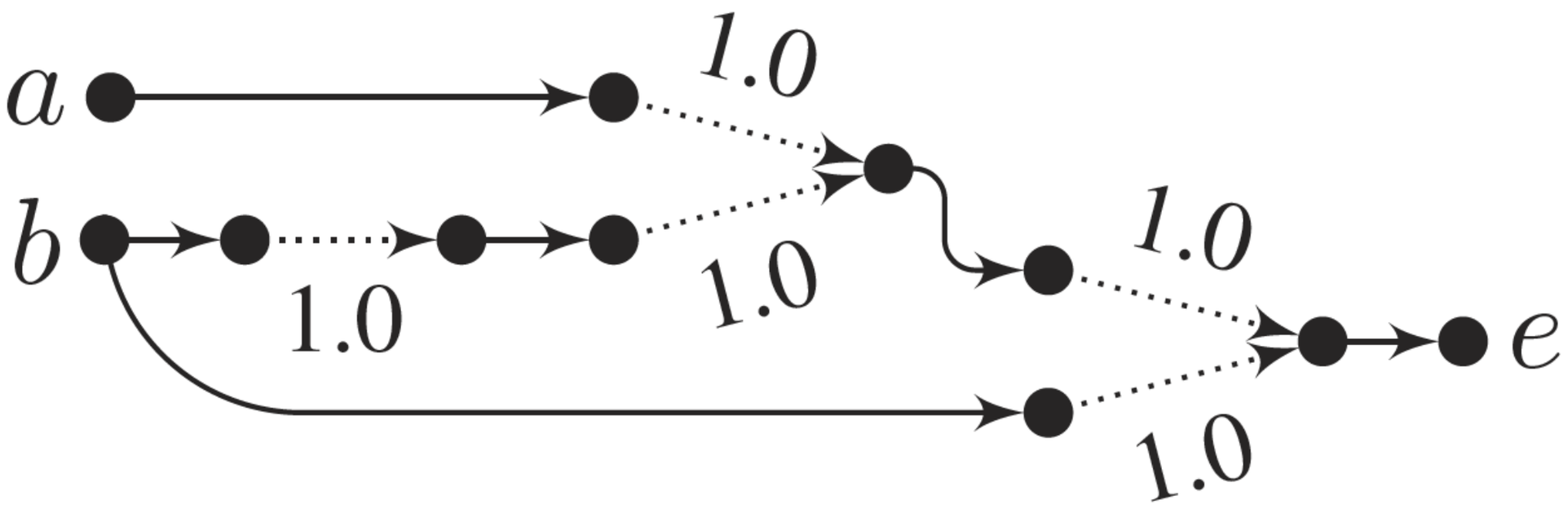
Timing Analysis



Timing Analysis

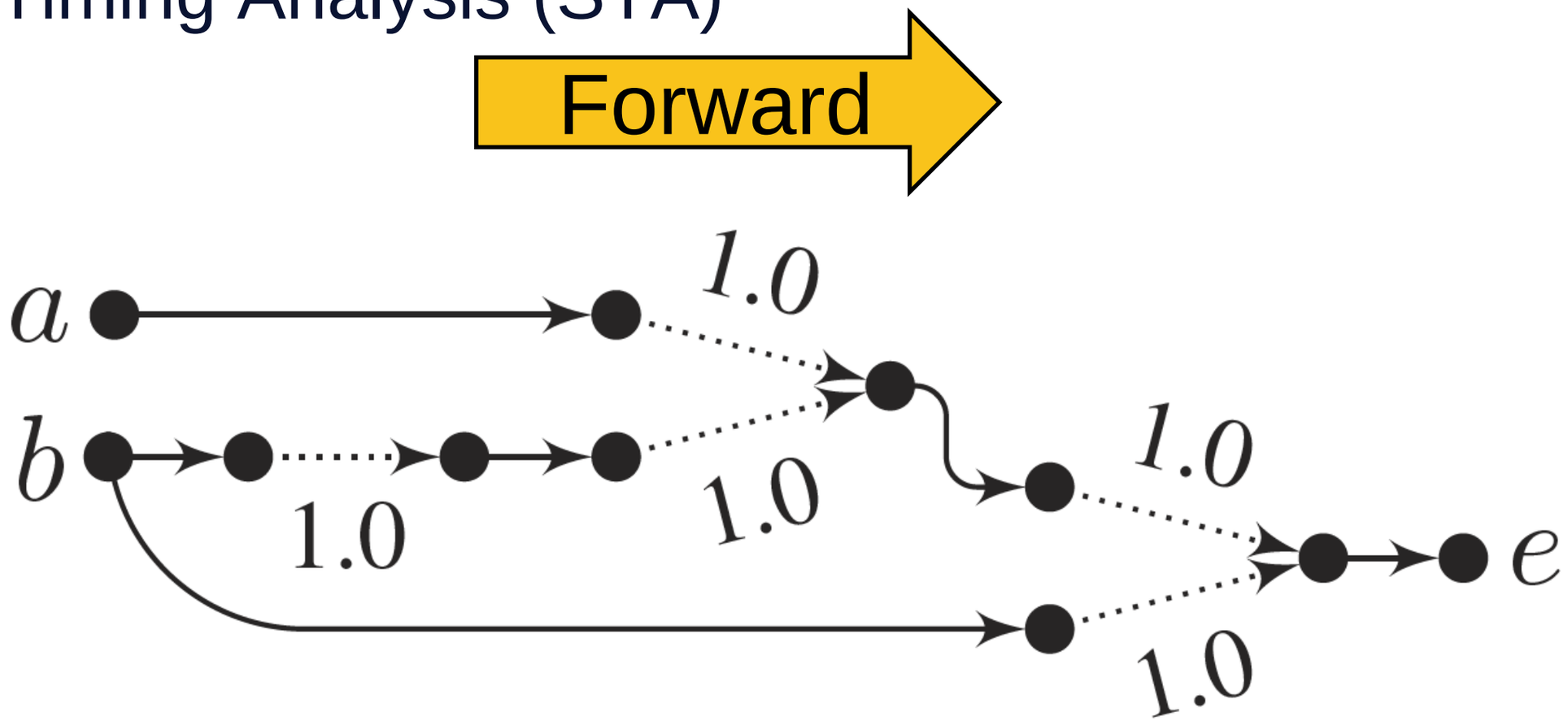


Static Timing Analysis (STA)



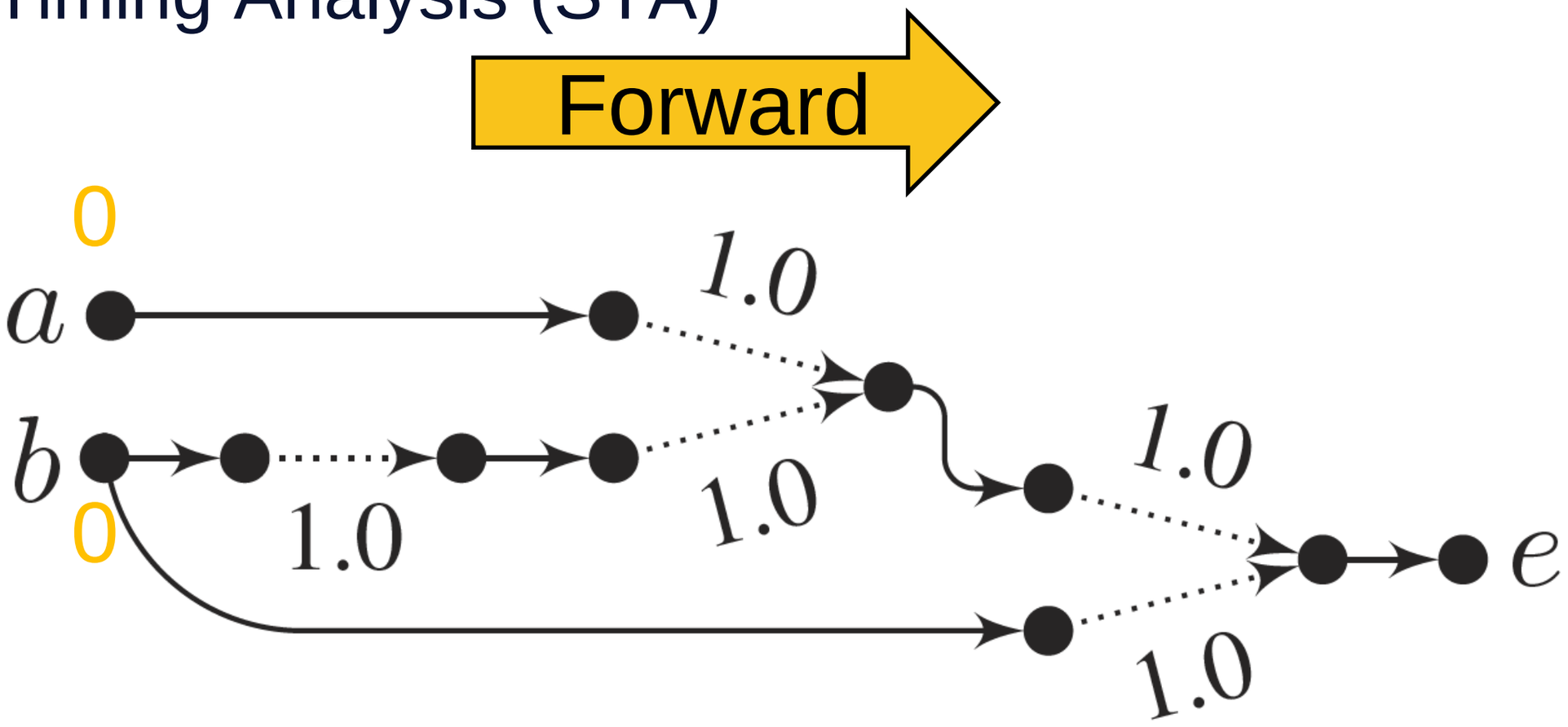
- Forward: Arrival Times
- Backward: Required Times & Slack
- Linear Time: **O(V+E)**

Static Timing Analysis (STA)



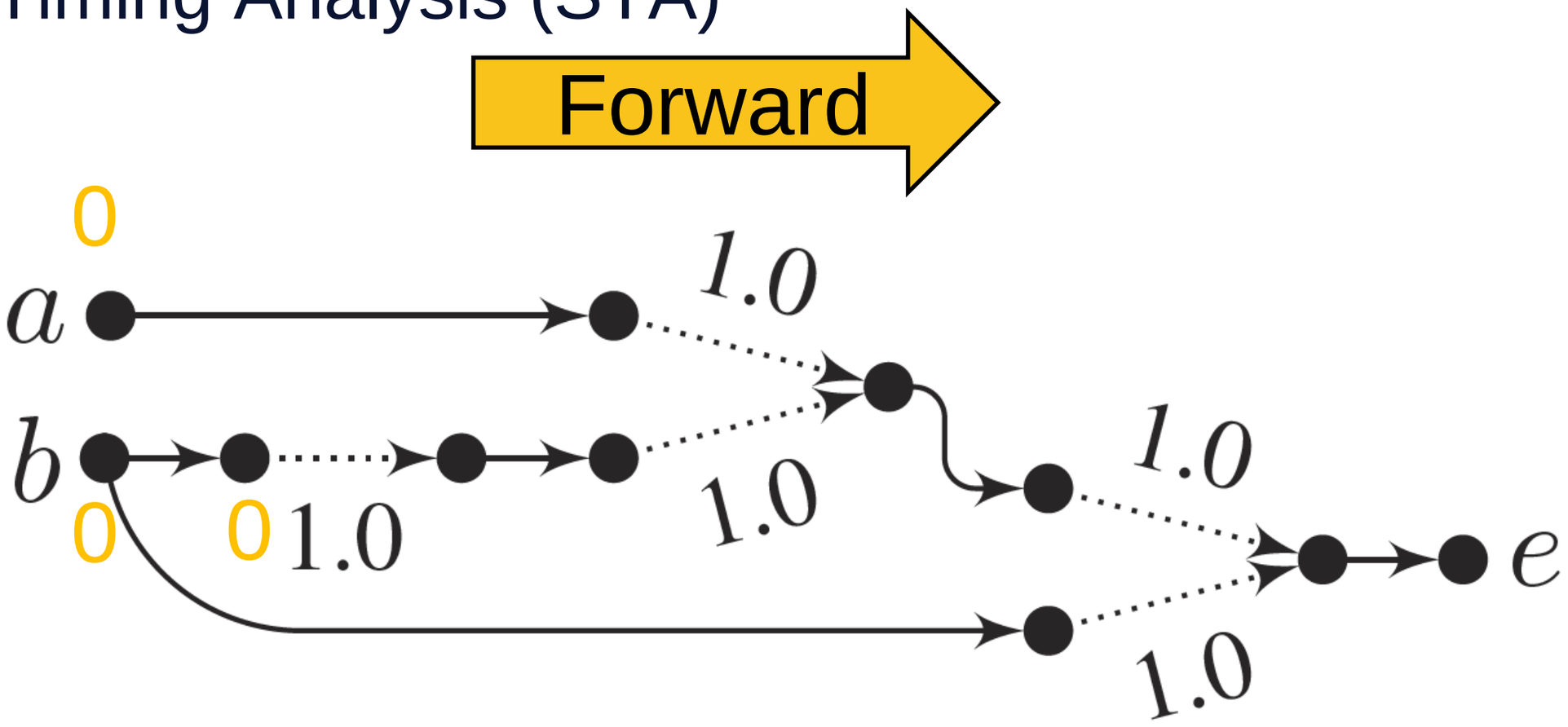
- Forward: Arrival Times
- Backward: Required Times & Slack
- Linear Time: **O(V+E)**

Static Timing Analysis (STA)



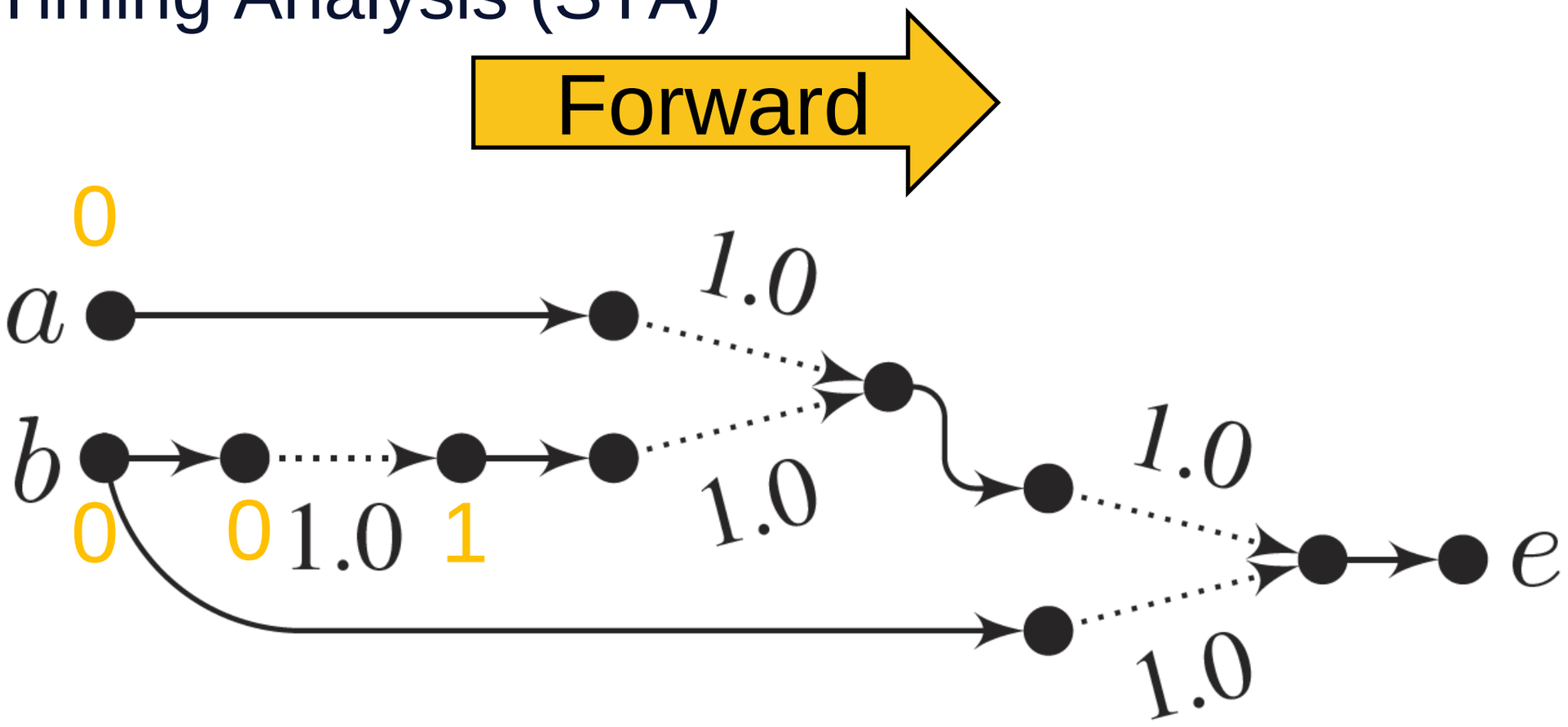
- Forward: Arrival Times
- Backward: Required Times & Slack
- Linear Time: **O(V+E)**

Static Timing Analysis (STA)



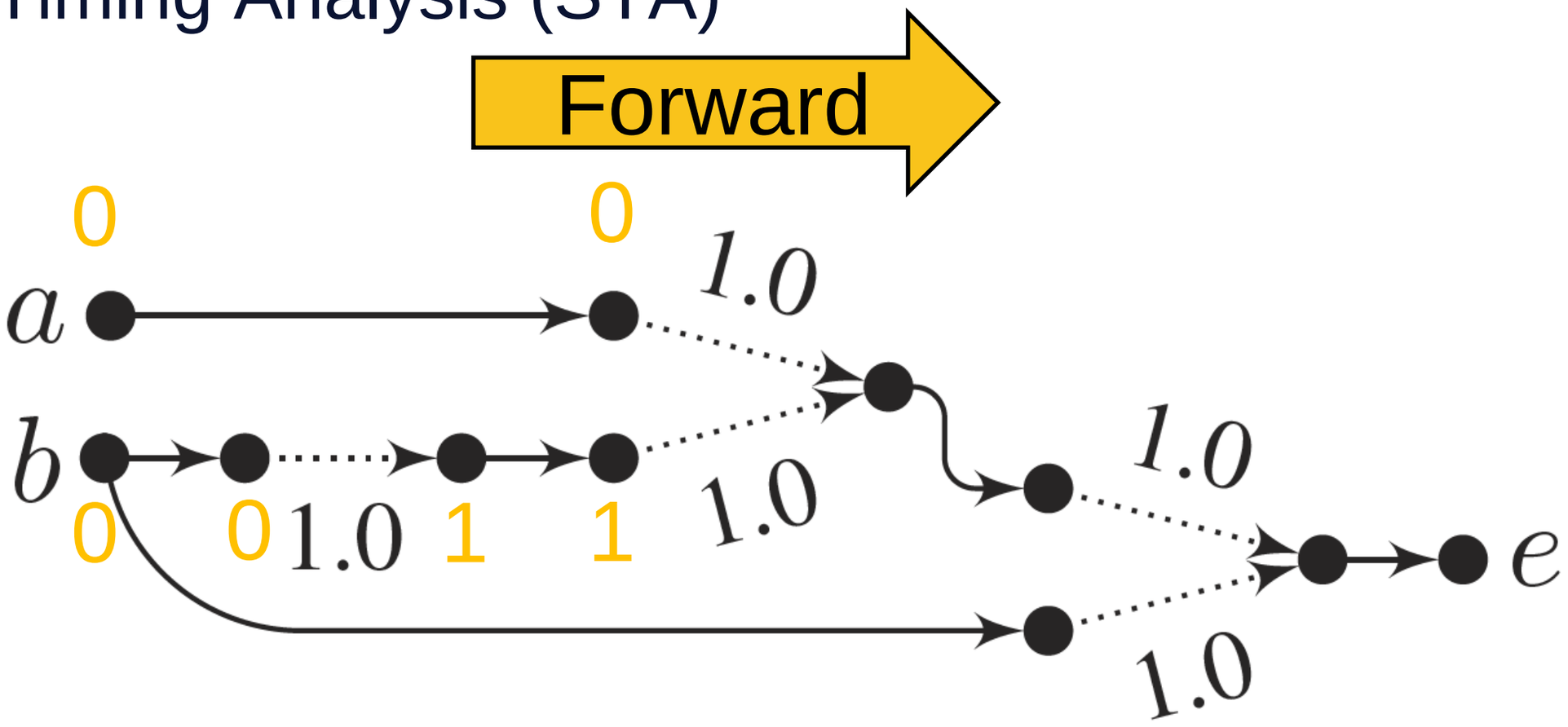
- Forward: Arrival Times
- Backward: Required Times & Slack
- Linear Time: **O(V+E)**

Static Timing Analysis (STA)



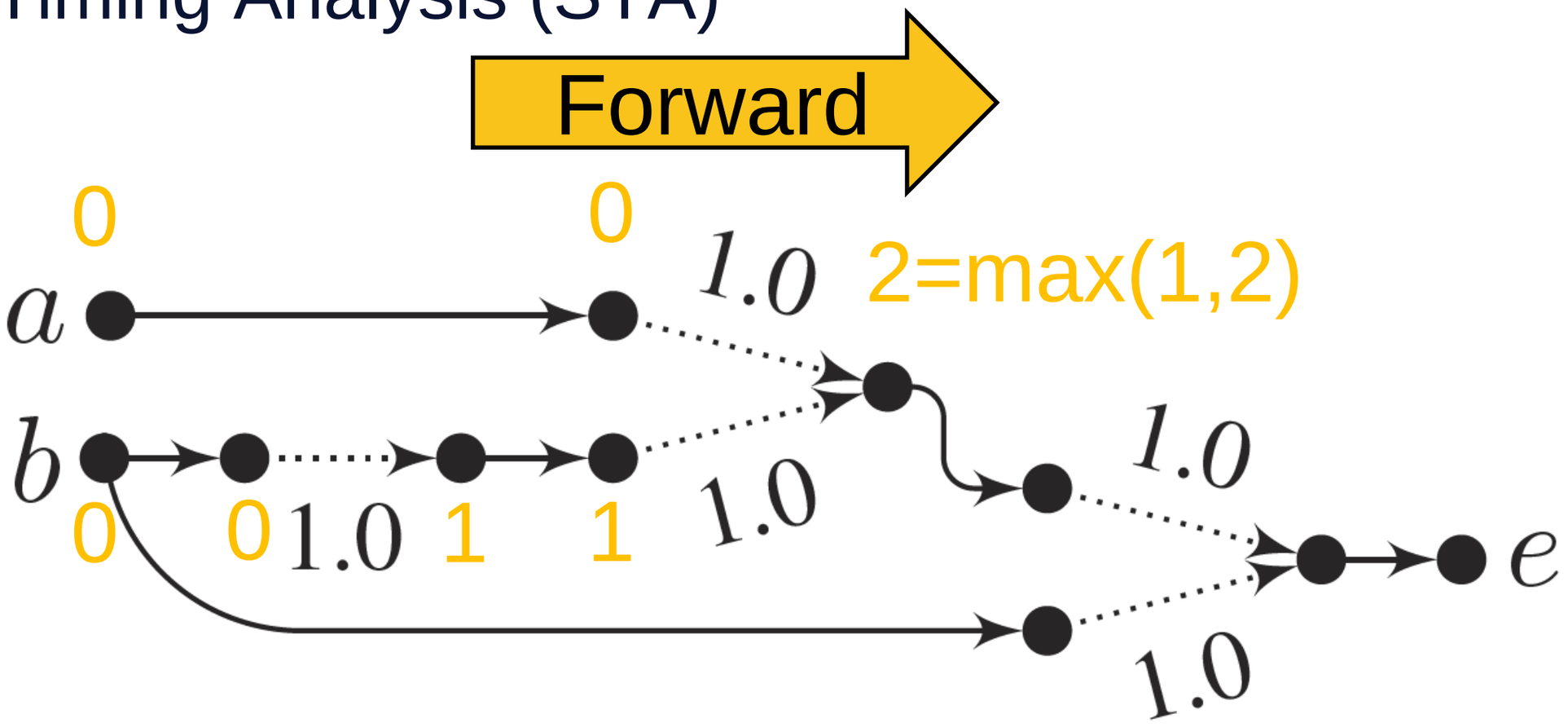
- Forward: Arrival Times
- Backward: Required Times & Slack
- Linear Time: **O(V+E)**

Static Timing Analysis (STA)



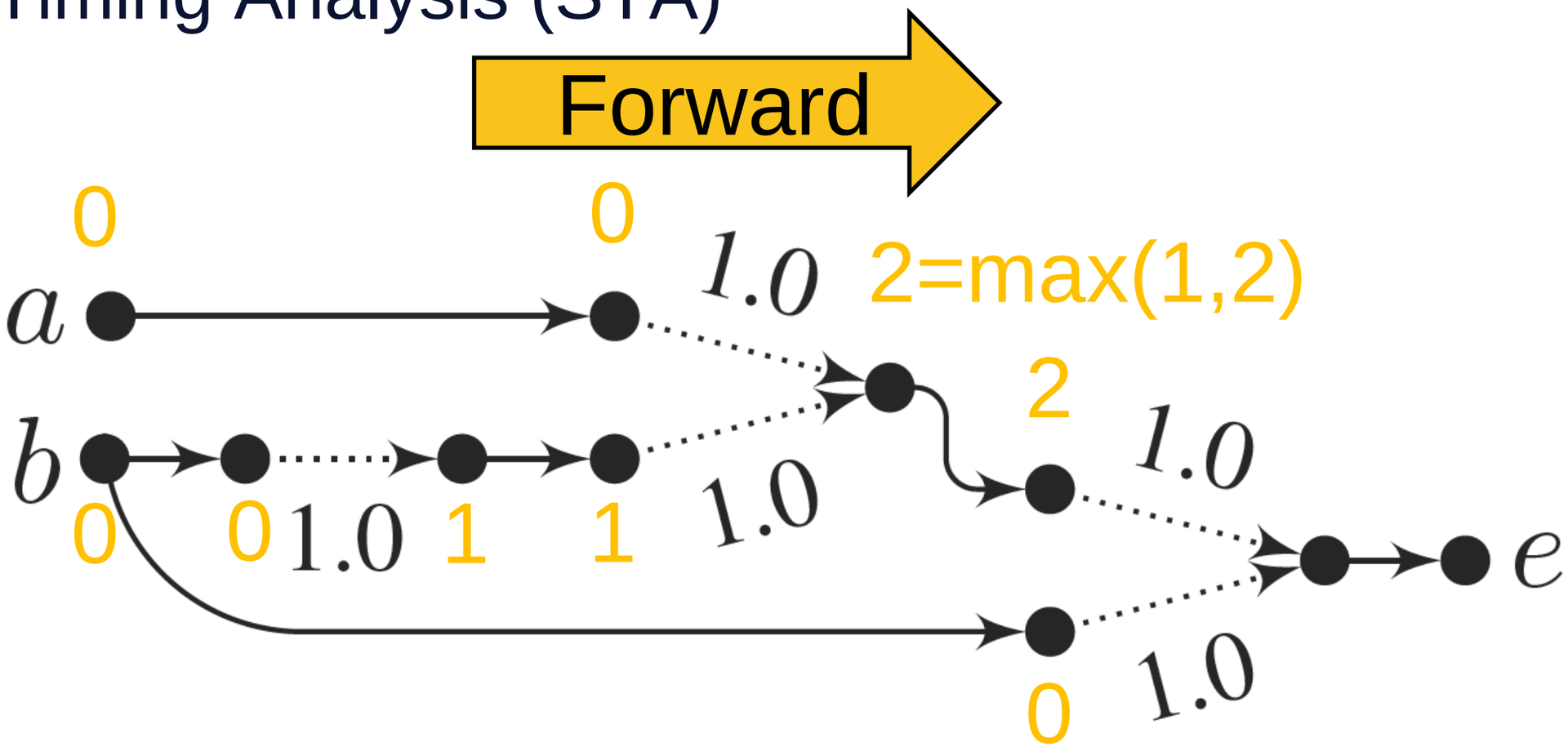
- Forward: Arrival Times
- Backward: Required Times & Slack
- Linear Time: **O(V+E)**

Static Timing Analysis (STA)



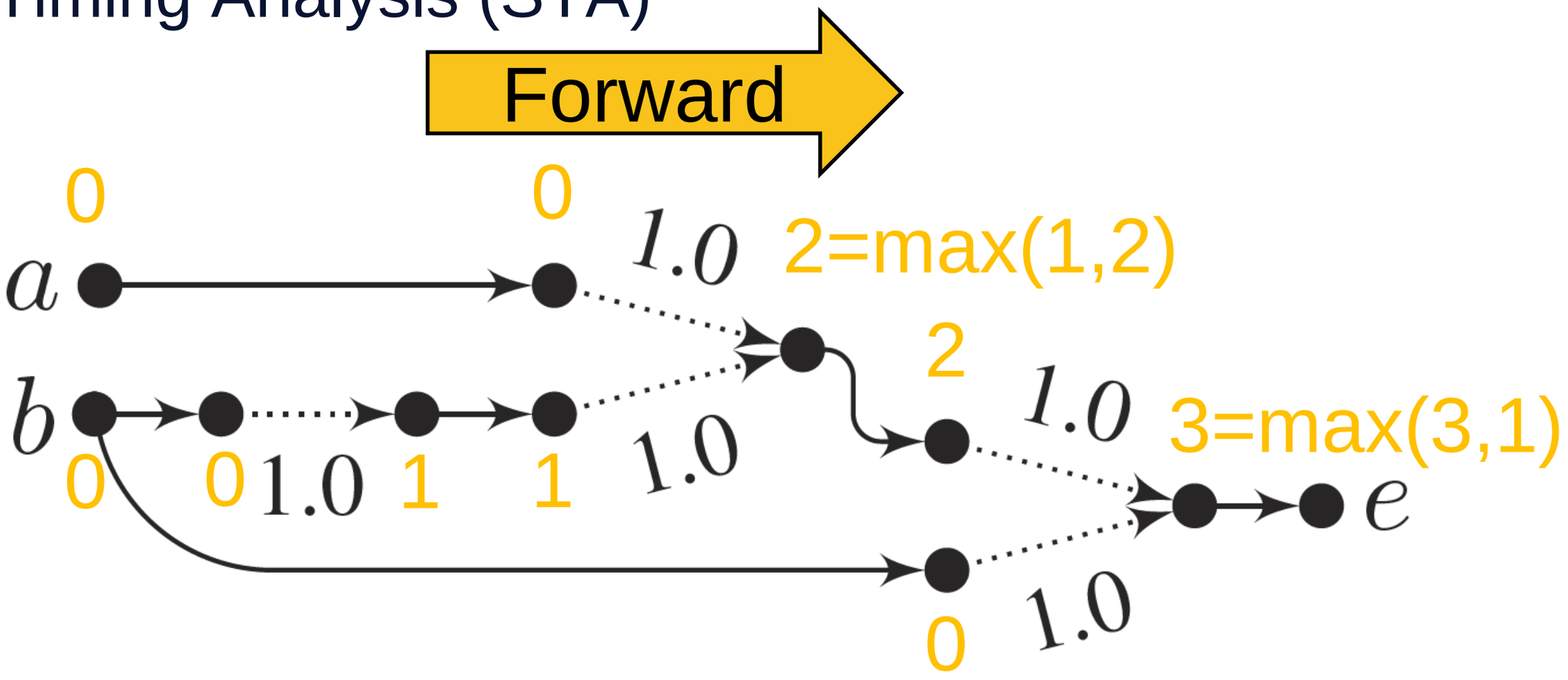
- Forward: Arrival Times
- Backward: Required Times & Slack
- Linear Time: **O(V+E)**

Static Timing Analysis (STA)



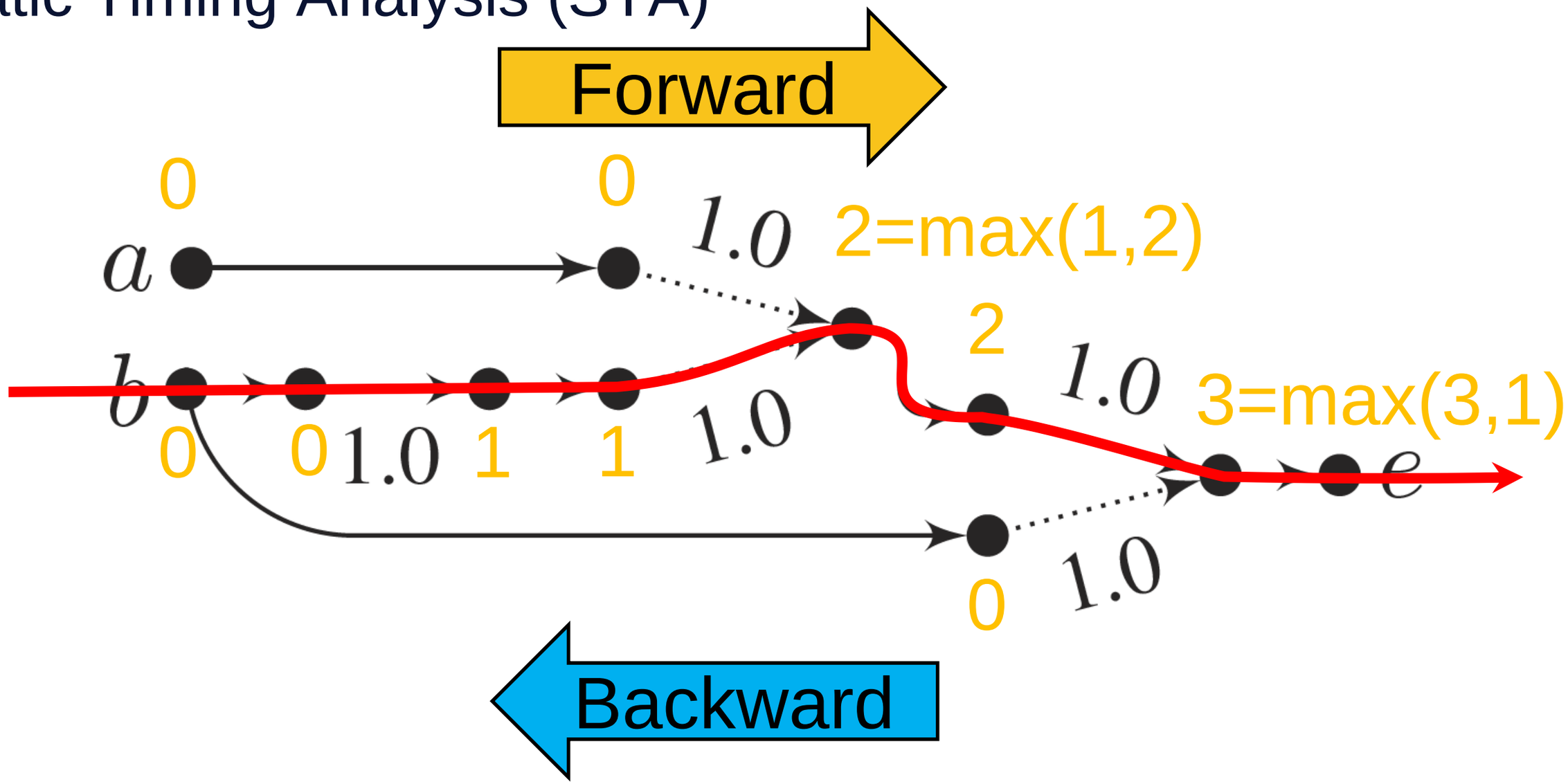
- Forward: Arrival Times
- Backward: Required Times & Slack
- Linear Time: **O(V+E)**

Static Timing Analysis (STA)



- Forward: Arrival Times
- Backward: Required Times & Slack
- Linear Time: **O(V+E)**

Static Timing Analysis (STA)



- Forward: Arrival Times
- Backward: Required Times & Slack
- Linear Time: **O(V+E)**

Tatum

Tatum

- High performance & full featured STA engine
- Supports:
  - Maximum (Setup) & Minimum (Hold) Analysis
  - Multiple Clocks
  - Various Timing Constraints/Exceptions
    - *False Paths*
    - *Multicycle Paths*
  - Application defined Timing Graph
  - Application defined Delay Calculator

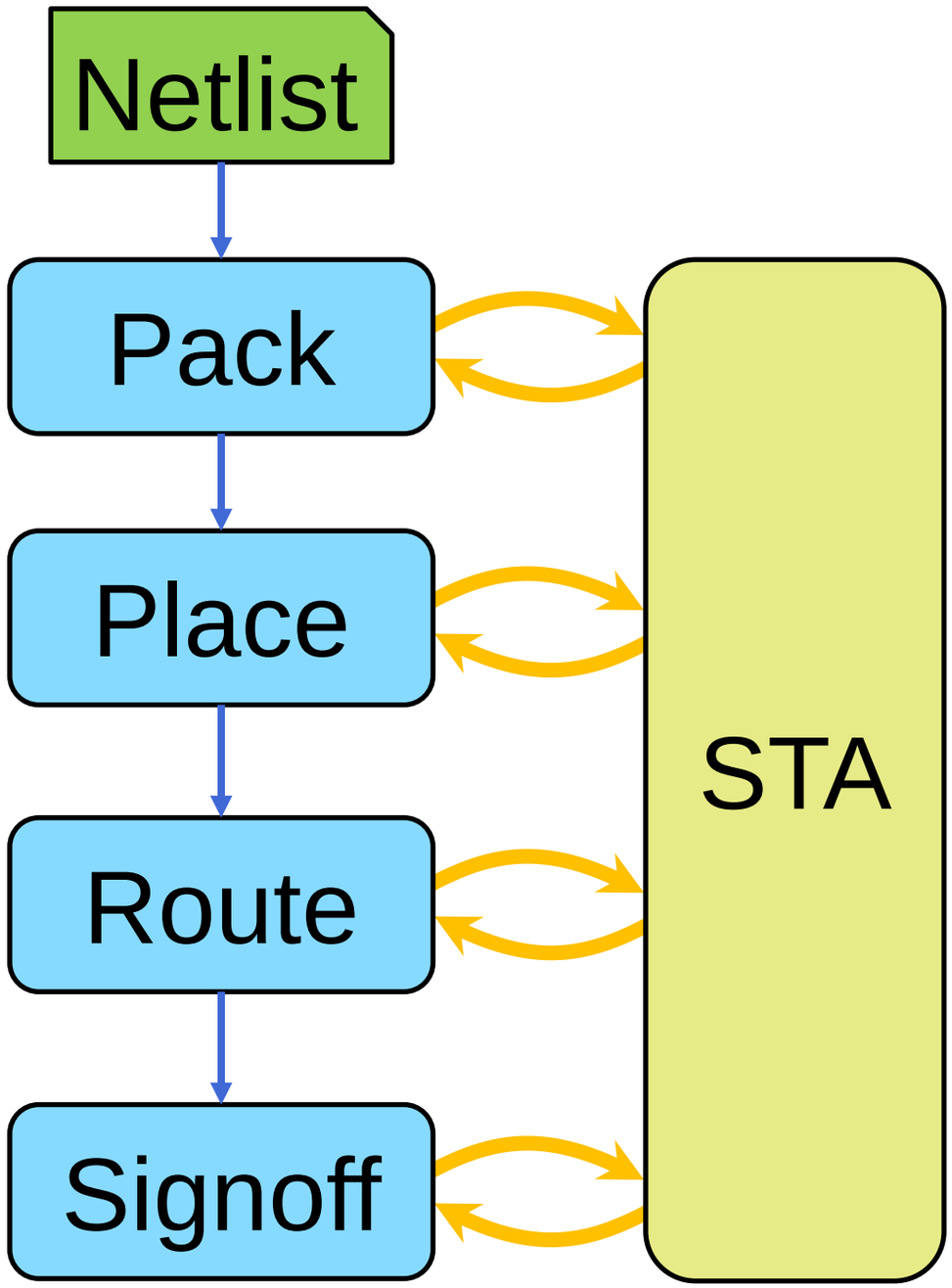
Tatum

- High performance & full featured STA engine
- Supports:
  - Maximum (Setup) & Minimum (Hold) Analysis
  - Multiple Clocks
  - Various Timing Constraints/Exceptions
    - *False Paths*
    - *Multicycle Paths*
  - Application defined Timing Graph
  - Application defined Delay Calculator

Not Enough!

FPGA Design Flow & STA

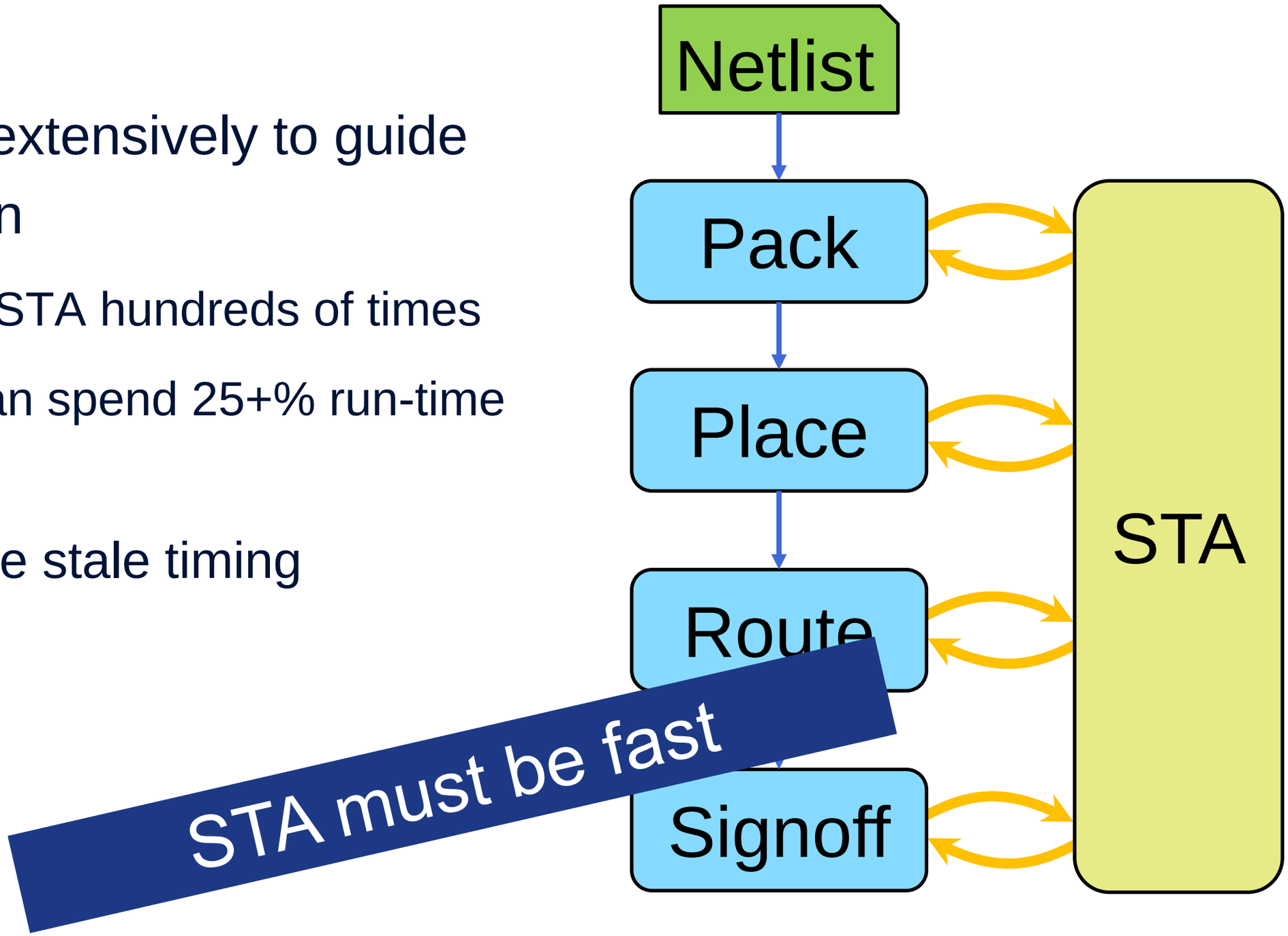
- STA used extensively to guide optimization
  - VPR calls STA hundreds of times
  - Quartus can spend 25+% run-time on STA [1]
- Typically use stale timing information



[1] M. Hutton et al., “Efficient static timing analysis and applications using edge masks,” FPGA, 2005

FPGA Design Flow & STA

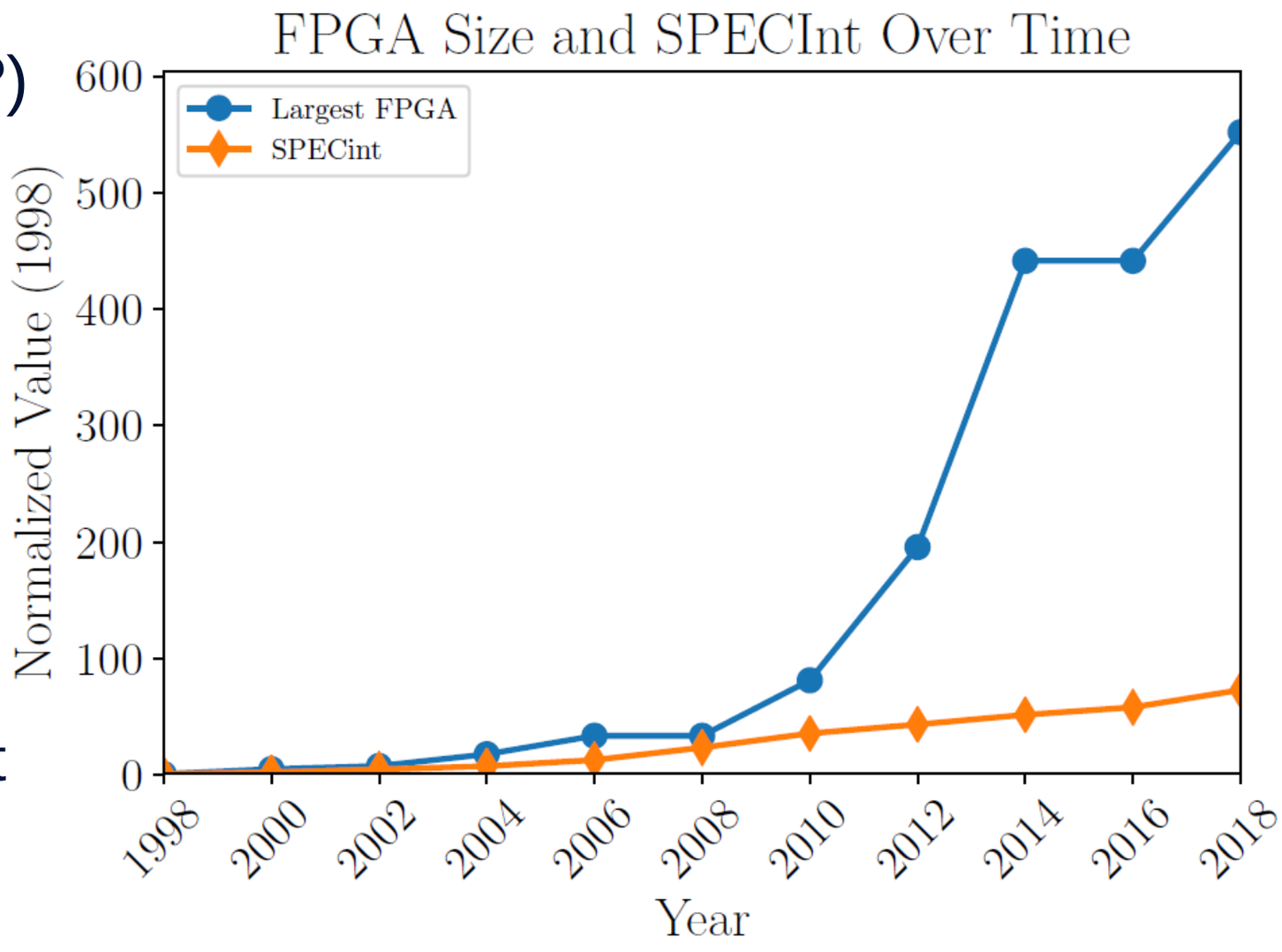
- STA used extensively to guide optimization
  - VPR calls STA hundreds of times
  - Quartus can spend 25+% run-time on STA [1]
- Typically use stale timing information



[1] M. Hutton et al., “Efficient static timing analysis and applications using edge masks,” FPGA, 2005

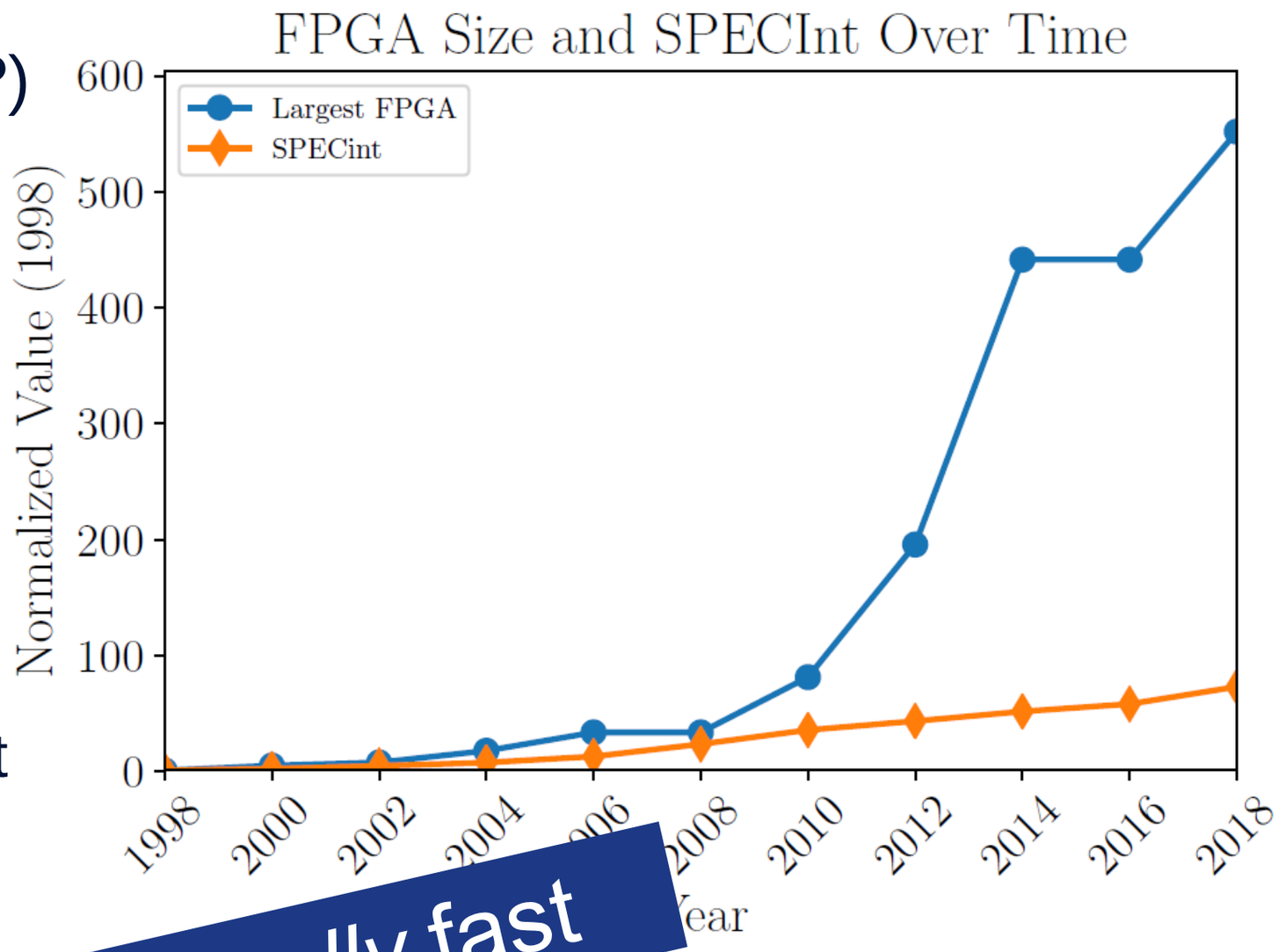
Design Flow Pressures on STA Run-Time

- More clock-domains (10s, 100s?)
- More timing corners (1, 2, 4, 8?)
- Increasing design size
- Limited single-threaded performance gains
- Increasing CAD flow parallelization
  - Ahmdal's law: 4x speed-up limit at 25% STA!



Design Flow Pressures on STA Run-Time

- More clock-domains (10s, 100s?)
- More timing corners (1, 2, 4, 8?)
- Increasing design size
- Limited single-threaded performance gains
- Increasing CAD flow parallelization
  - Ahmdal's law: 4x speed-up limit at 25% STA!



STA must be really fast

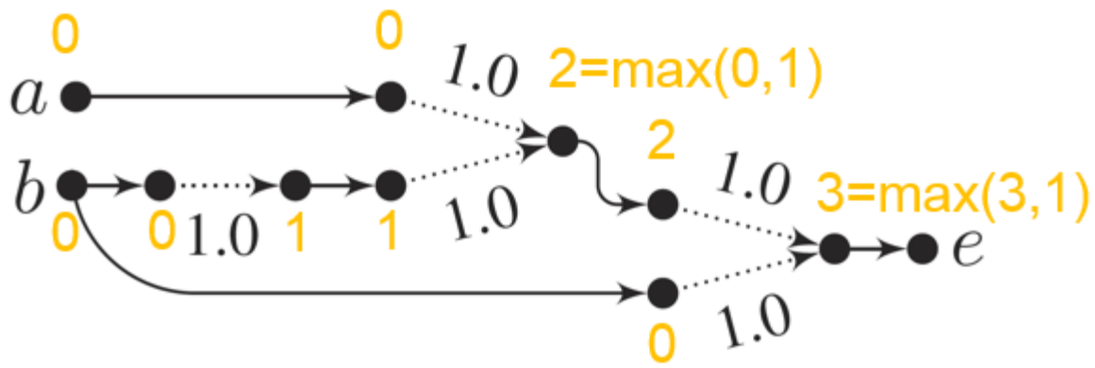
How to Speed-Up STA?

Algorithmic Improvements?

- **Bad News:** Already linear time!

STA has low Arithmetic Intensity:

- Lots of memory access (edges, nodes, times)
- Limited computation (SUM, MAX)
- Try to maximize data re-use!
  - *Efficient data structures*
  - *Single set of traversals for:*
    - *All clock domains & timing corners*
    - *Setup & Hold Analysis*



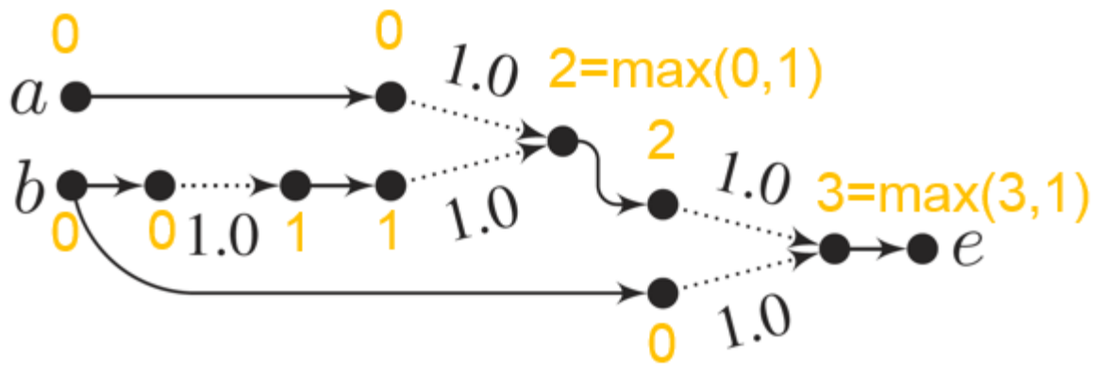
How to Speed-Up STA?

Algorithmic Improvements?

- **Bad News:** Already linear time!

STA has low Arithmetic Intensity:

- Lots of memory access (edges, nodes, times)
- Limited computation (SUM, MAX)
- Try to maximize data re-use!
  - *Efficient data structures*
  - *Single set of traversals for:*
    - *All clock domains & timing corners*
    - *Setup & Hold Analysis*

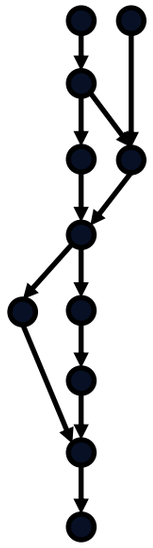


Parallel STA?

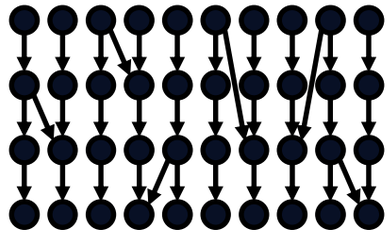
STA Parallelization

STA Parallelization Challenges

- Dependency Structure
  - Timing Graph defines dependencies
  - Must be respected in parallelization



High Logic Depth



Deeply Pipelined

- Low Arithmetic Intensity

STA Parallelization Challenges

- Dependency Structure
  - Timing Graph defines dependencies
  - Must be respected in parallelization

- Low Arithmetic Intensity

STA Parallelization Challenges

- Dependency Structure
  - Timing Graph defines dependencies
  - Must be respected in parallelization

- Low Arithmetic Intensity

Low Overhead Parallelization

Parallel Platforms

GPU?

CPU?

GPU Parallelization

	Speed-Up
Kernel	6.2x
Overall	0.9x

GPU Parallelization

	Speed-Up
Kernel	6.2x
Overall	0.9x

Data Transfer Limits GPU Speed-up!

GPU Parallelization

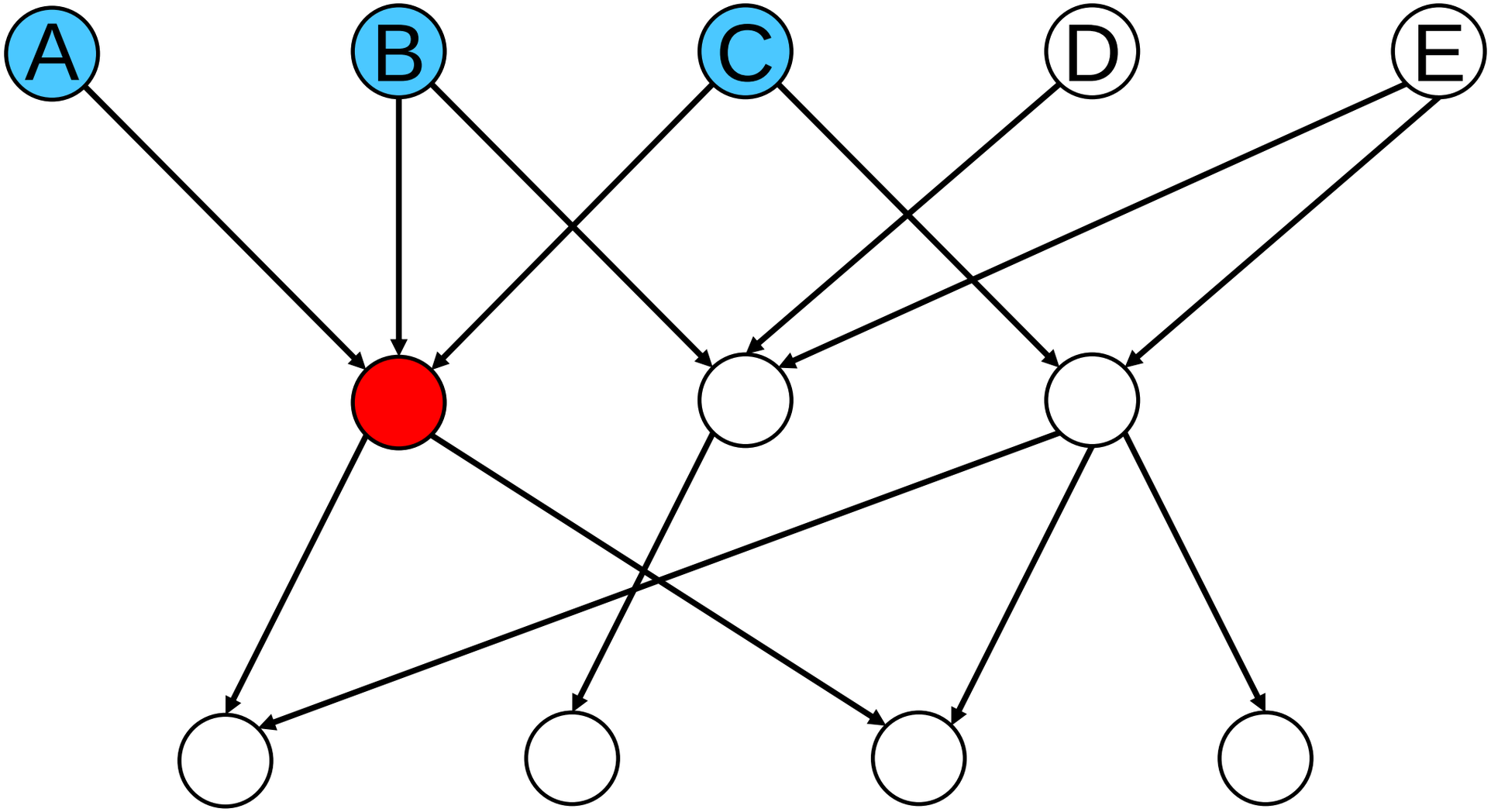
	Speed-Up
Kernel	6.2x
Overall	0.9x

Data Transfer Limits GPU Speed-up!

- Compute: O(V+E)
- Data: O(V+E)

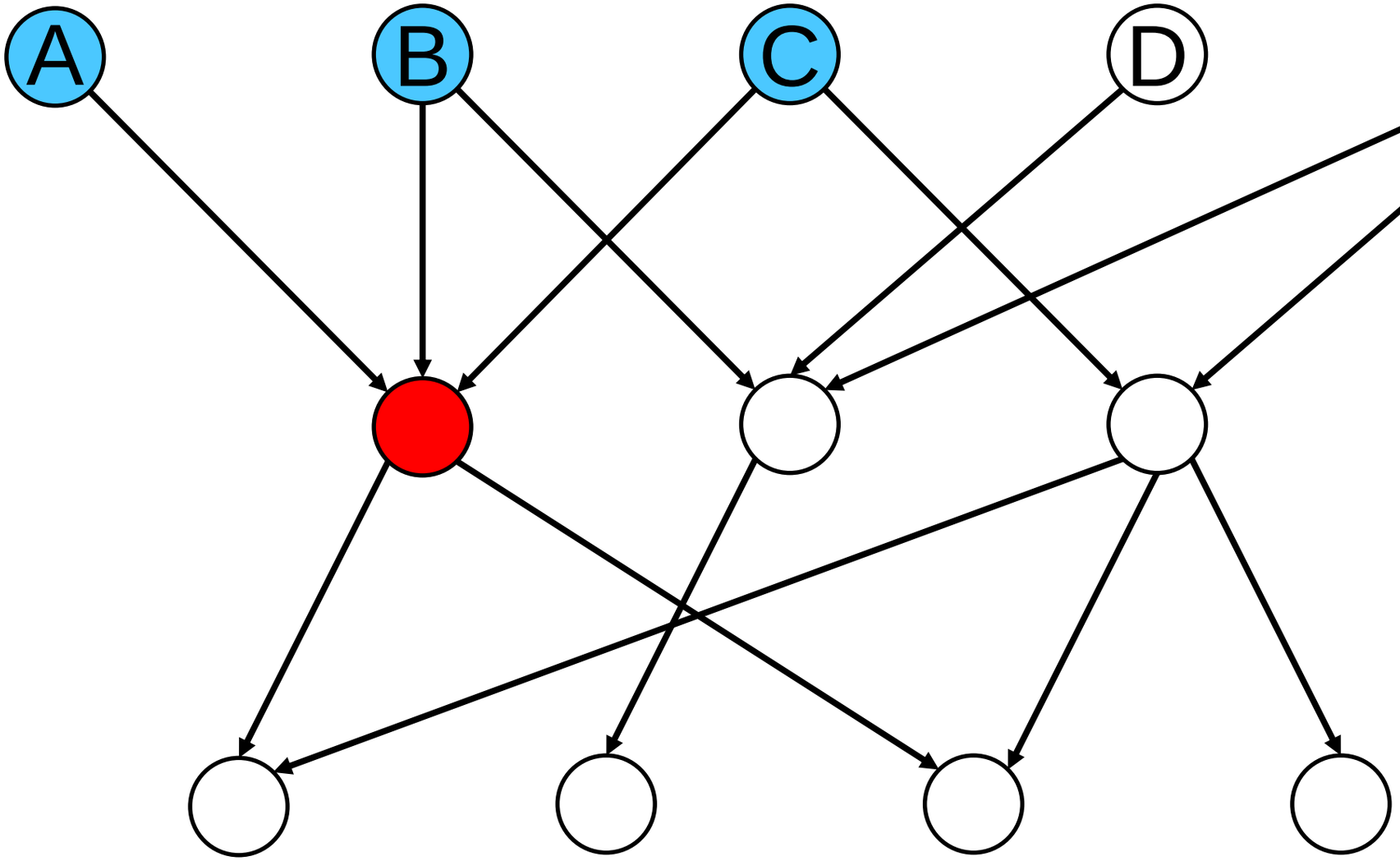
CPU Parallelization I

- **Push** Arrival Time from Source to Sink



CPU Parallelization I

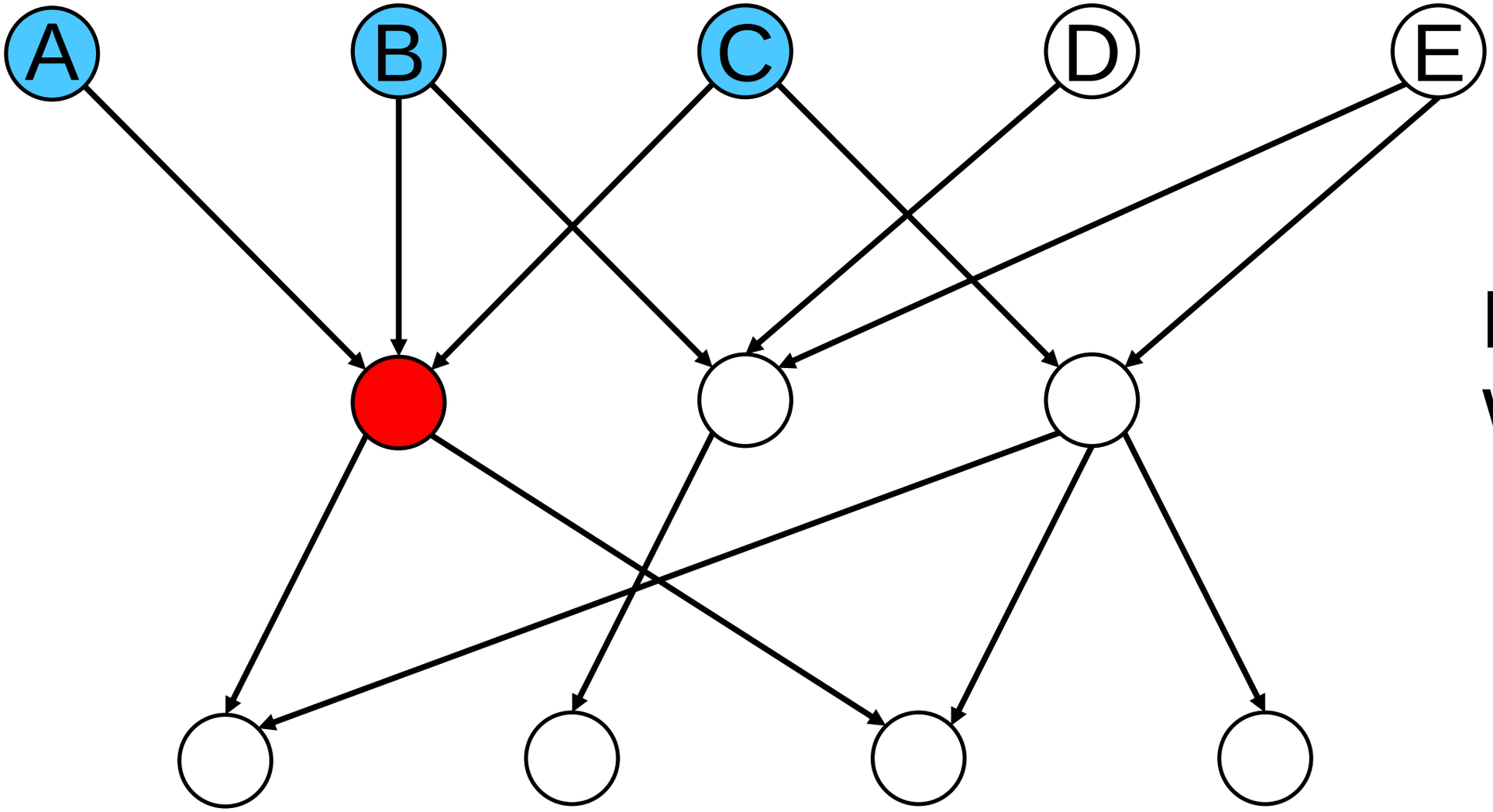
- **Push** Arrival Time from Source to Sink



Multiple Writers!

CPU Parallelization I

- **Push** Arrival Time from Source to Sink

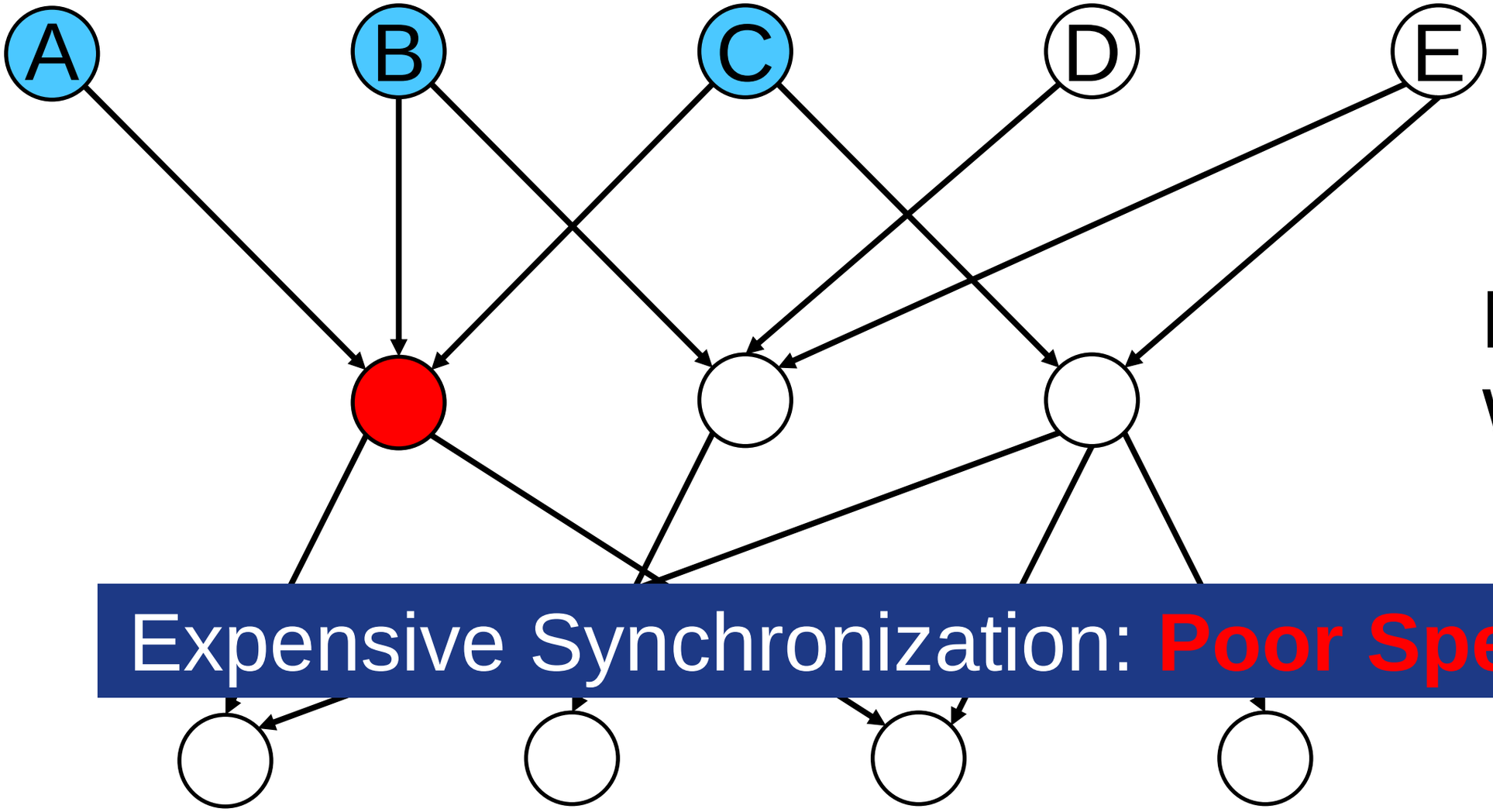


Multiple Writers!

Must use locks to synchronize

CPU Parallelization I

- **Push** Arrival Time from Source to Sink



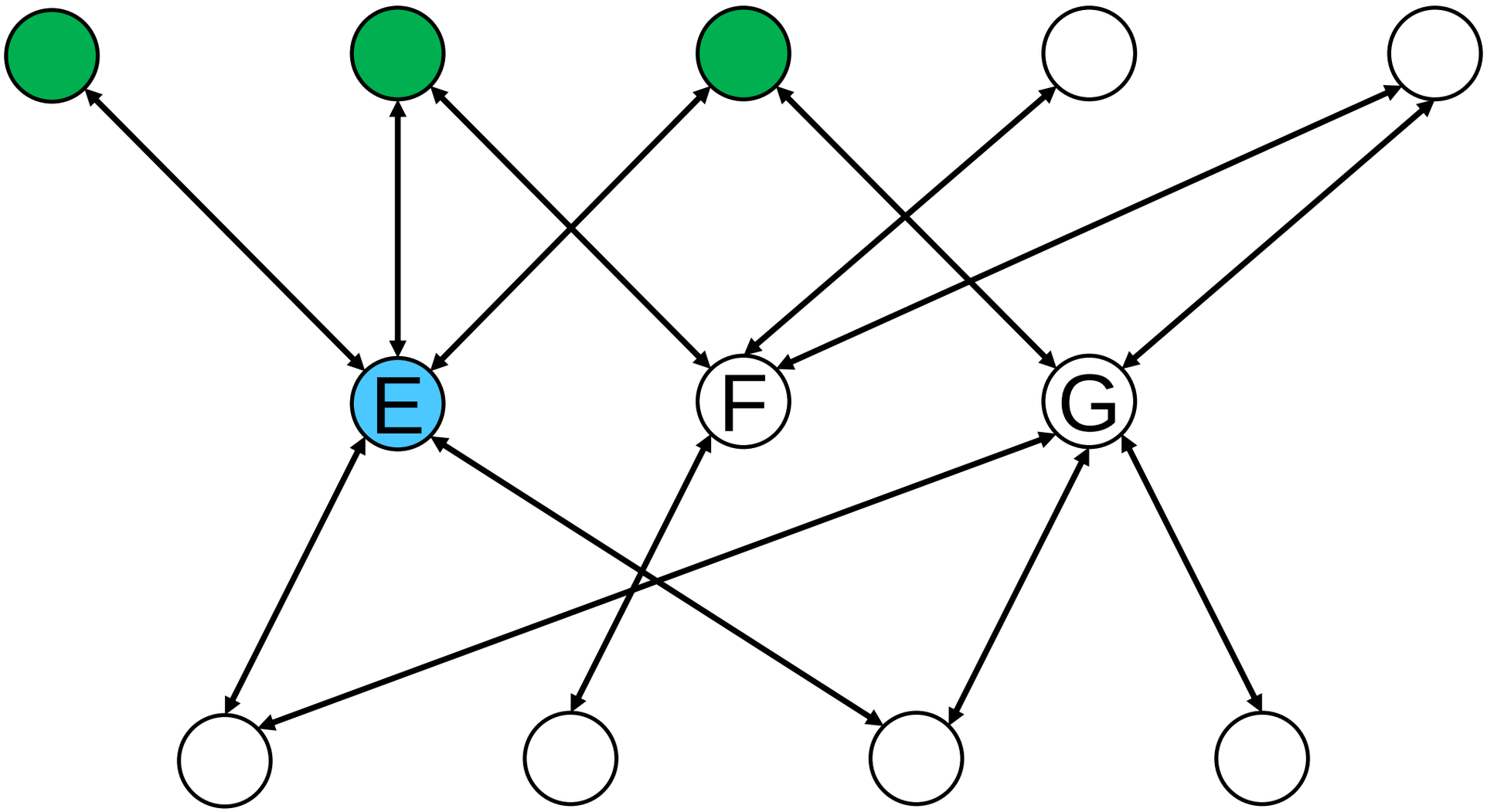
Multiple Writers!

Expensive Synchronization: **Poor Speed-Up**

Must use locks to synchronize

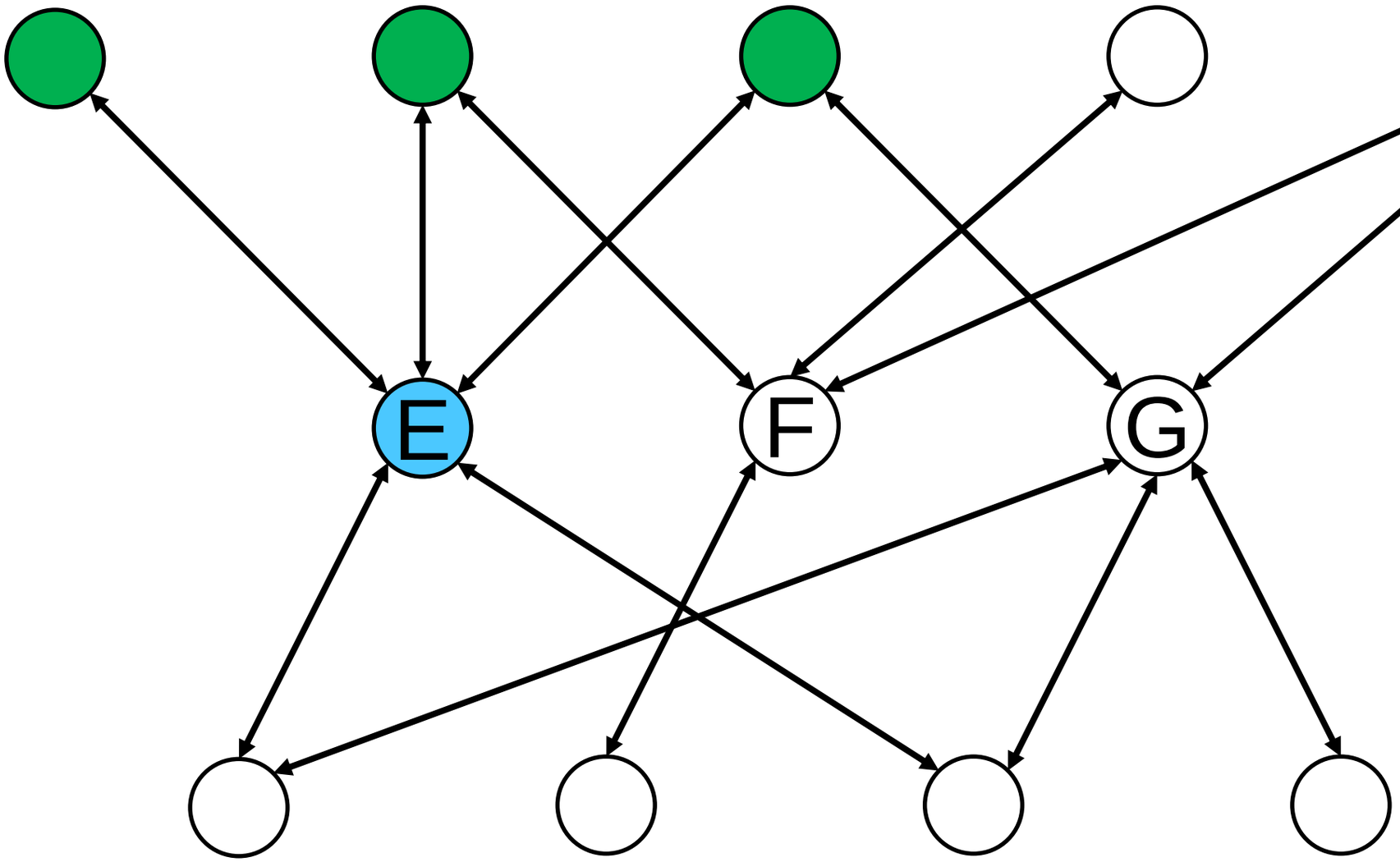
CPU Parallelization II

- **Pull** arrival times from Sources
- Requires bi-directional edges



CPU Parallelization II

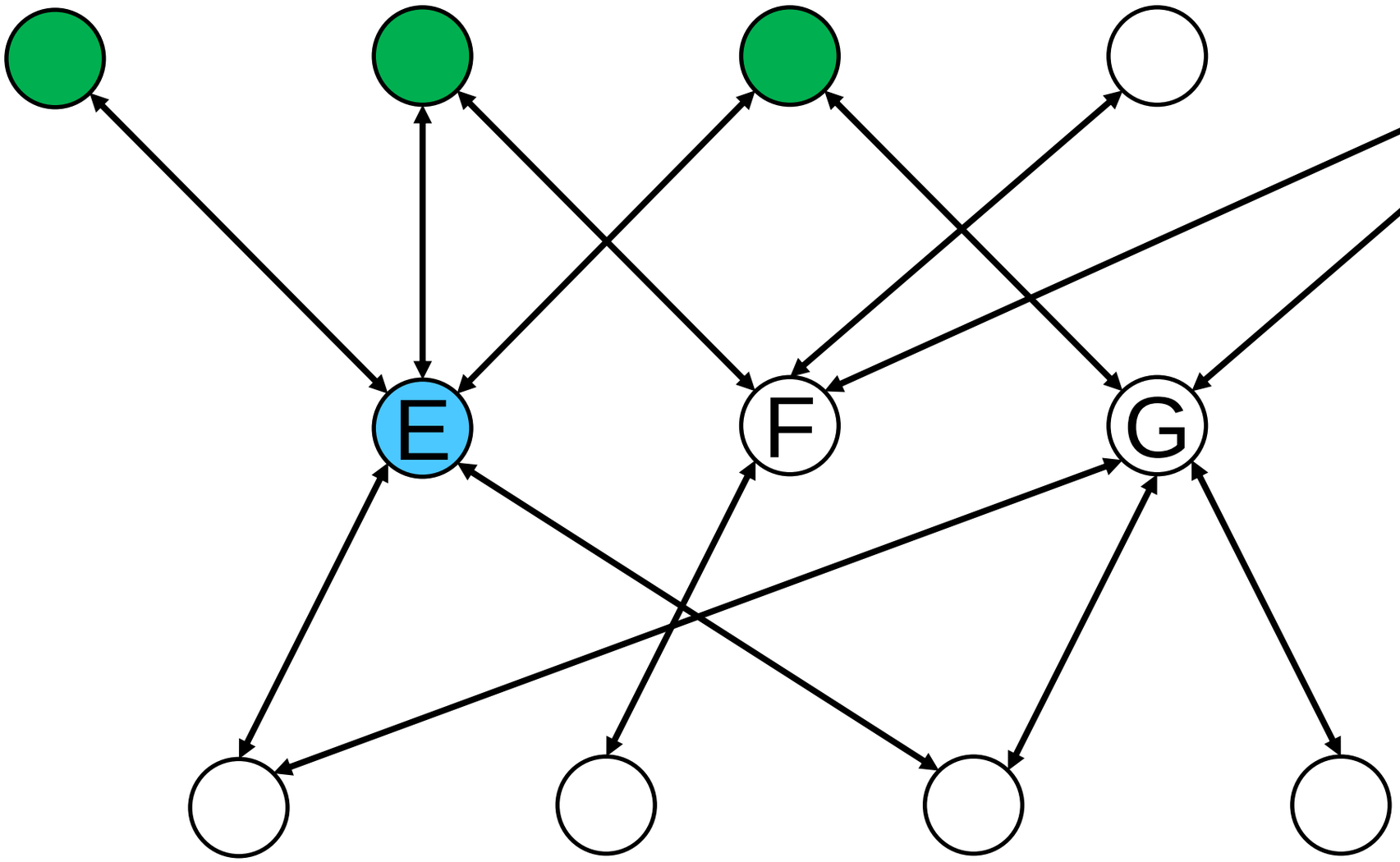
- **Pull** arrival times from Sources
- Requires bi-directional edges



Single Writer

CPU Parallelization II

- **Pull** arrival times from Sources
- Requires bi-directional edges

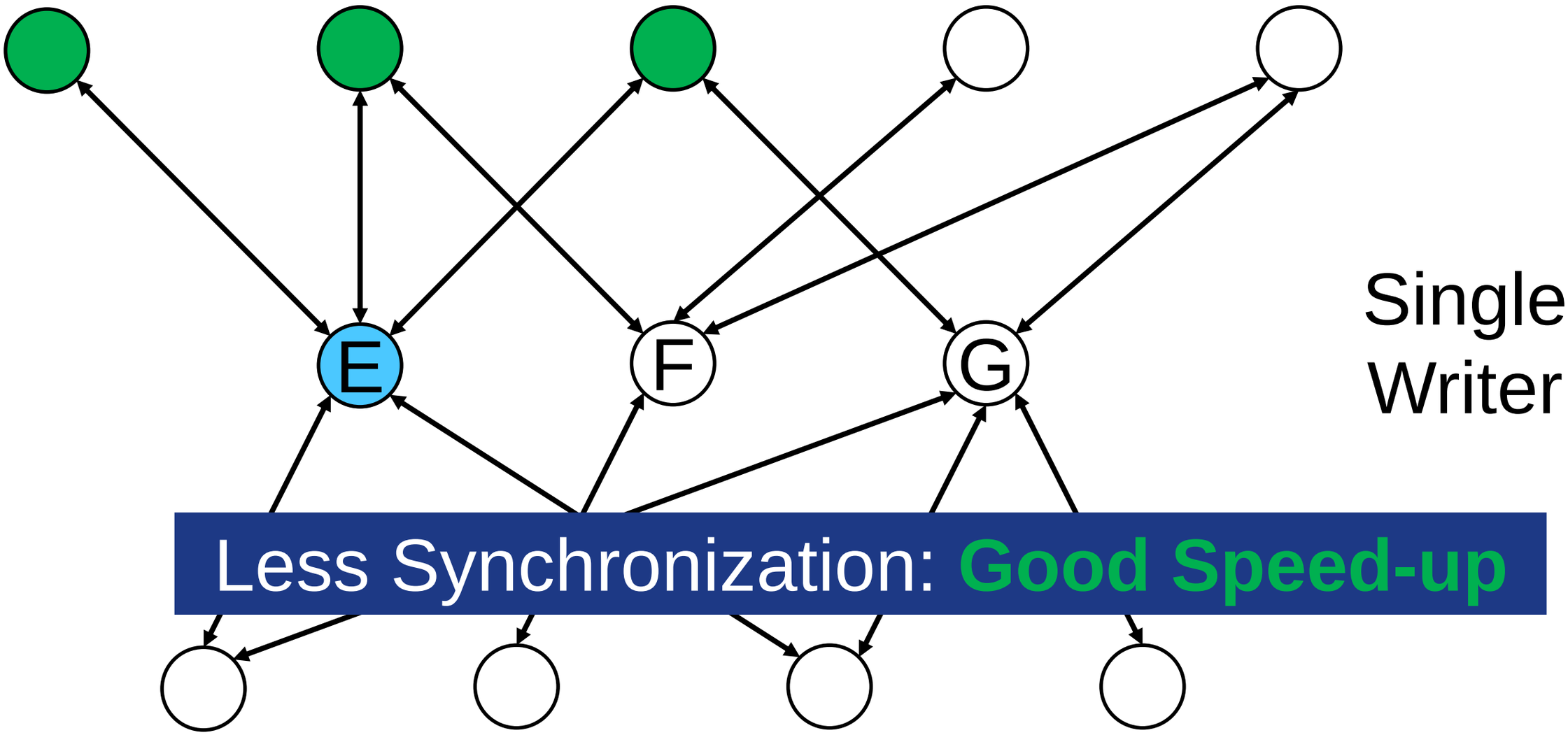


Single Writer

No Locks!

CPU Parallelization II

- **Pull** arrival times from Sources
- Requires bi-directional edges

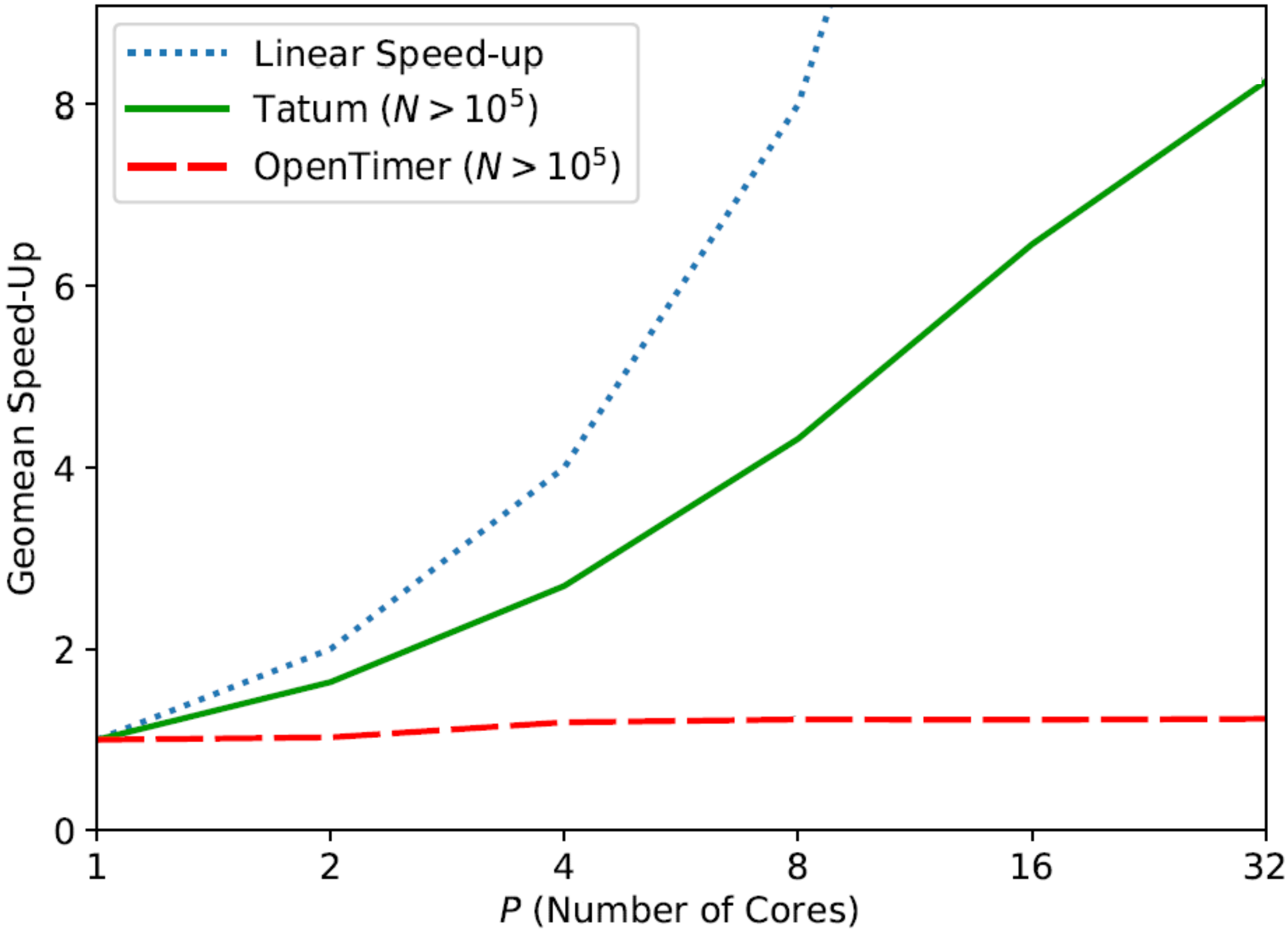


No Locks!

Performance Results

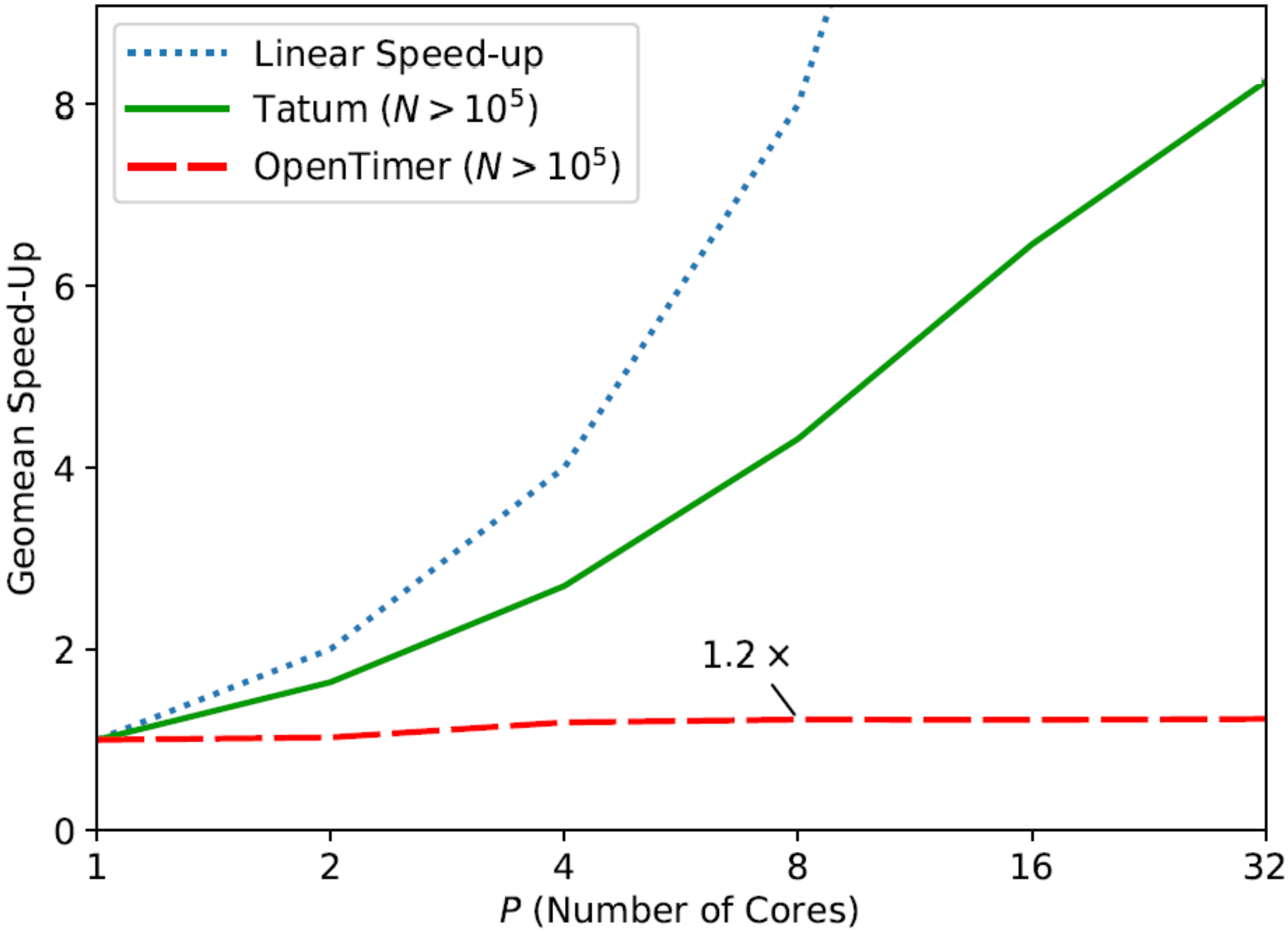
Tatum STA Performance: Tau'15 ASIC Benchmarks

- Self-relative speed-up



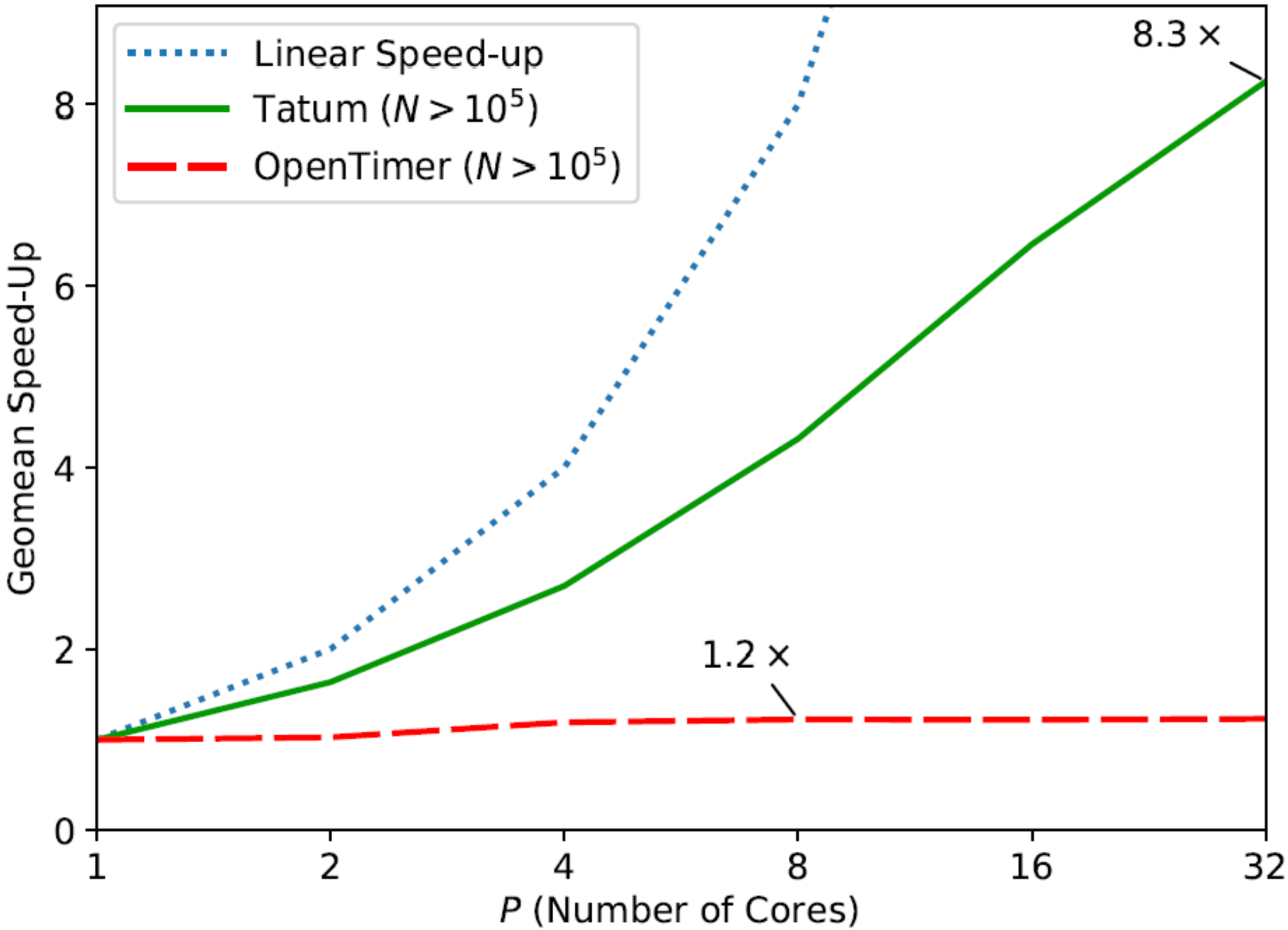
Tatum STA Performance: Tau'15 ASIC Benchmarks

- Self-relative speed-up



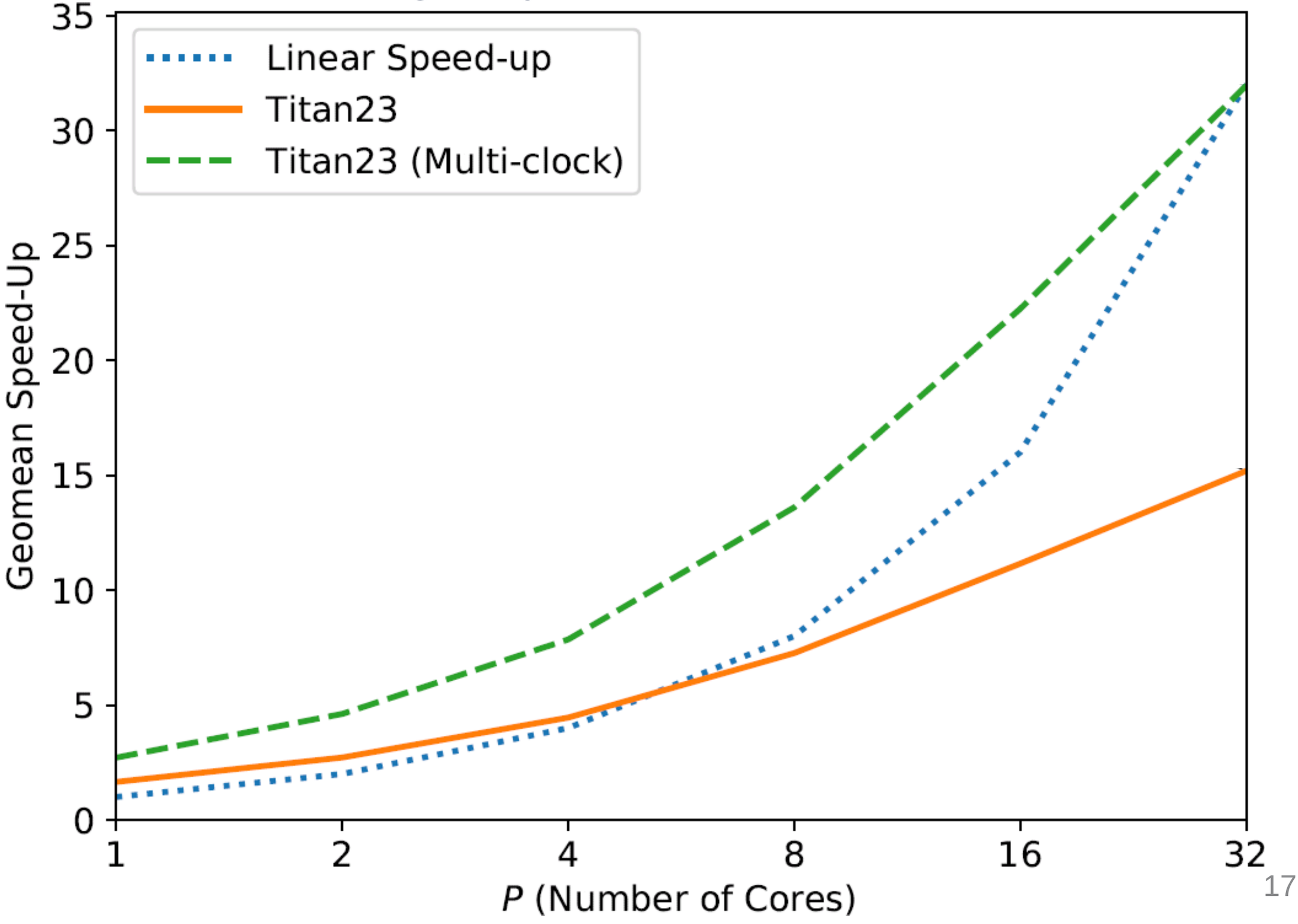
Tatum STA Performance: Tau'15 ASIC Benchmarks

- Self-relative speed-up



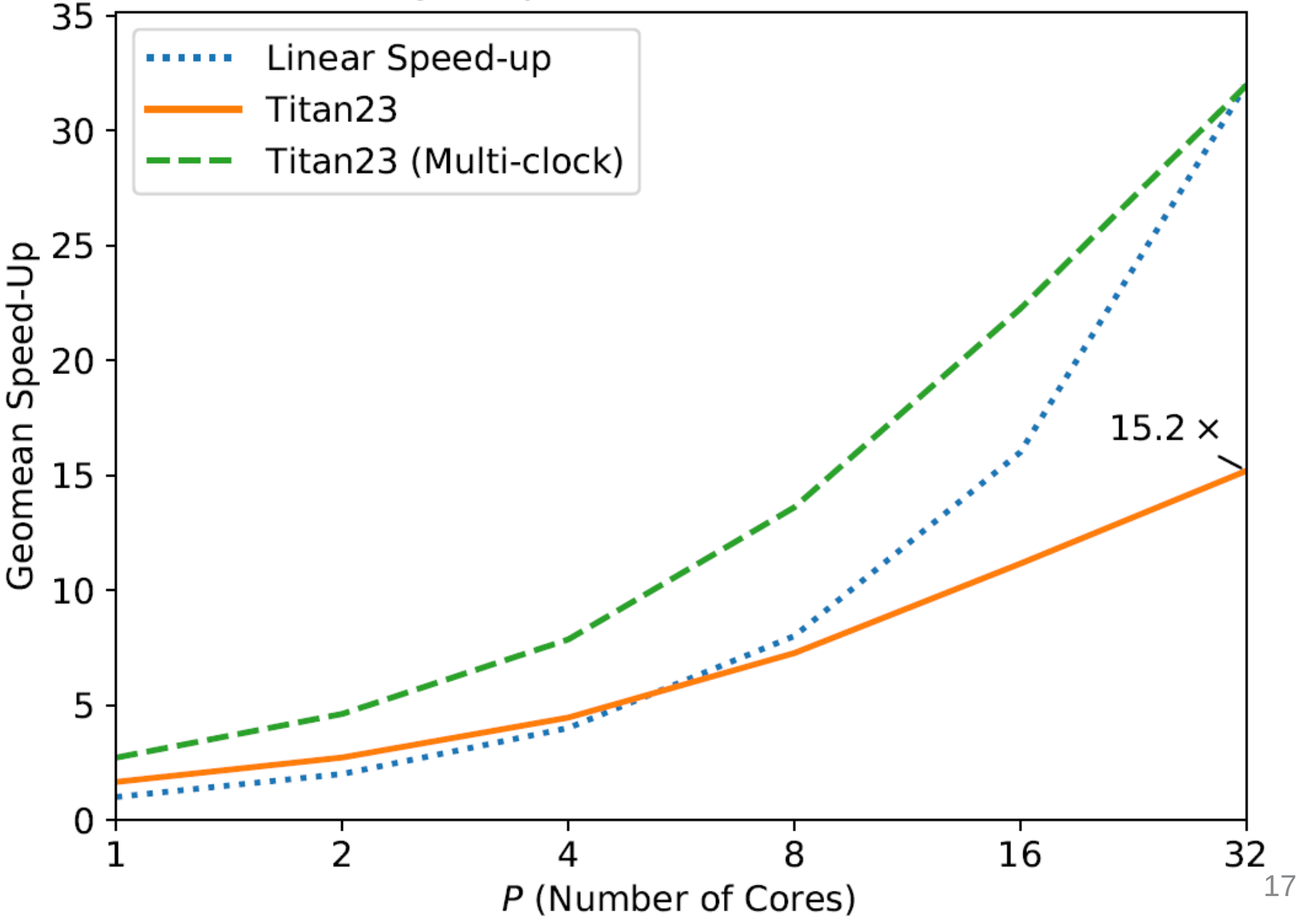
Tatum STA Performance: Titan Benchmarks

- Relative to VPR 7 timing analyzer



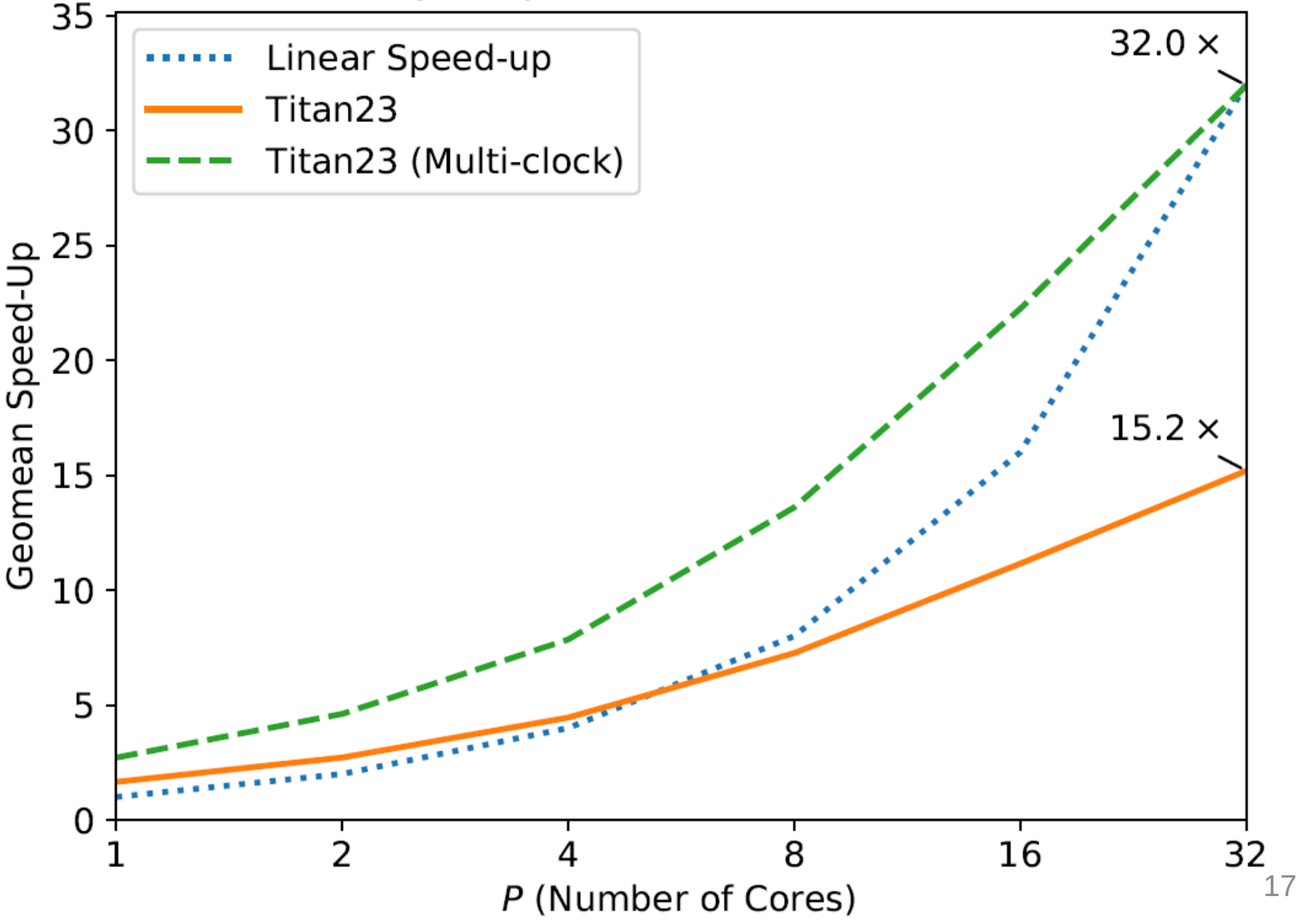
Tatum STA Performance: Titan Benchmarks

- Relative to VPR 7 timing analyzer



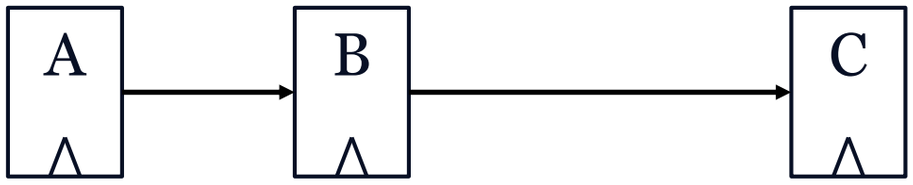
Tatum STA Performance: Titan Benchmarks

- Relative to VPR 7 timing analyzer



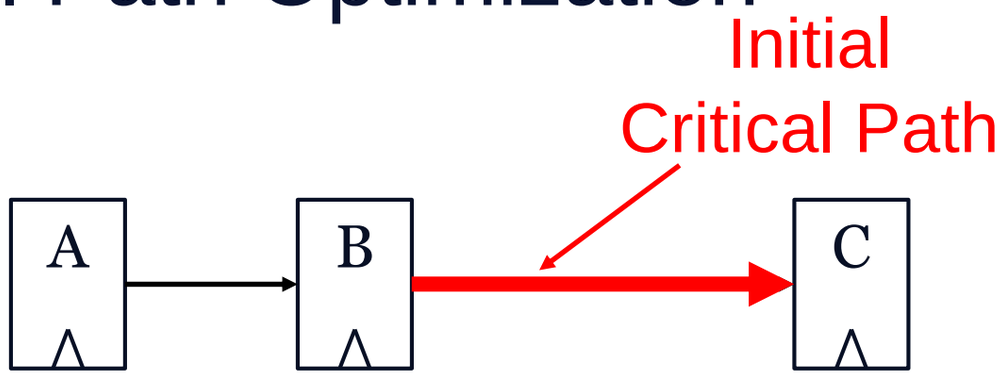
Improving Optimization

Placement Critical Path Optimization



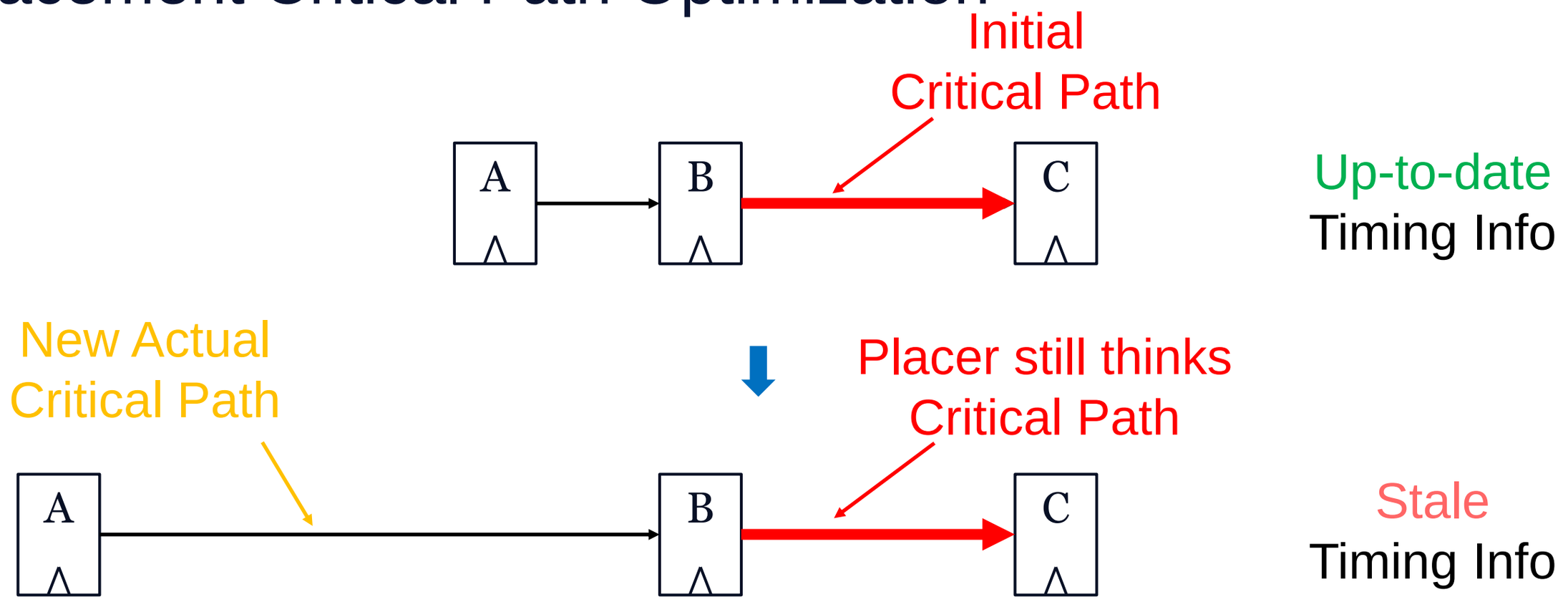
Up-to-date
Timing Info

Placement Critical Path Optimization

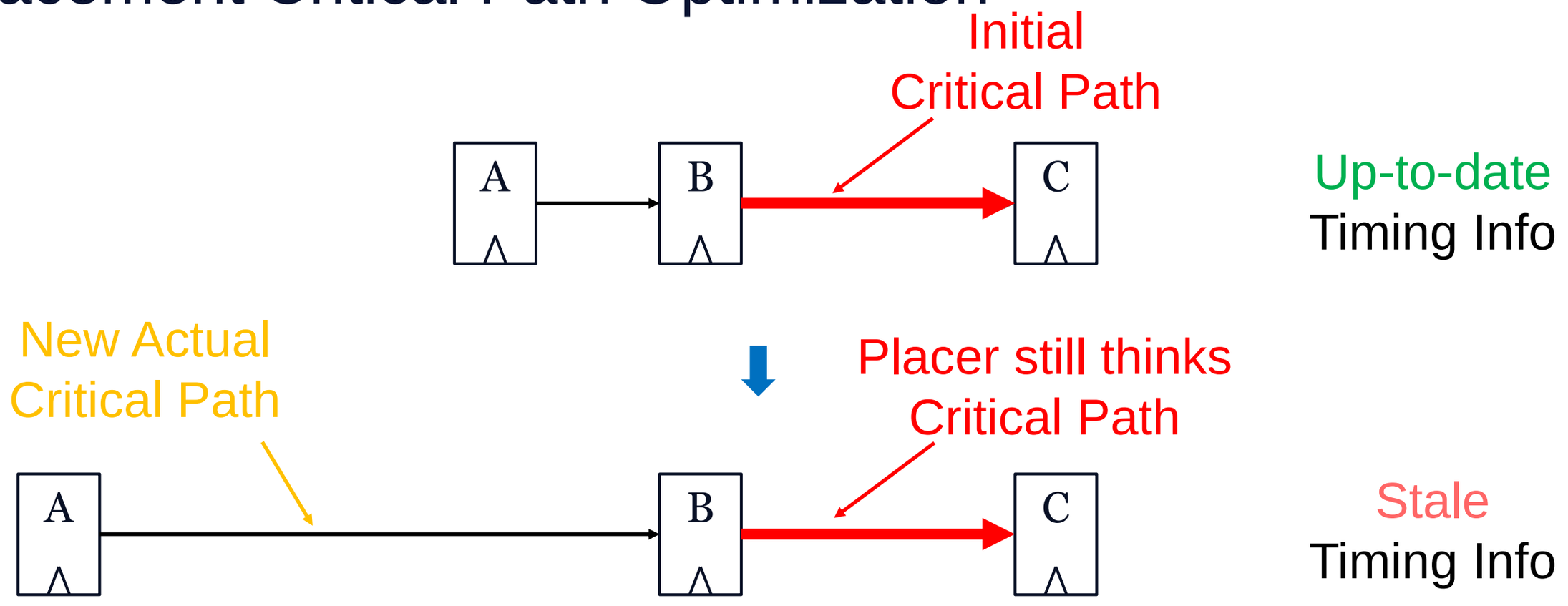


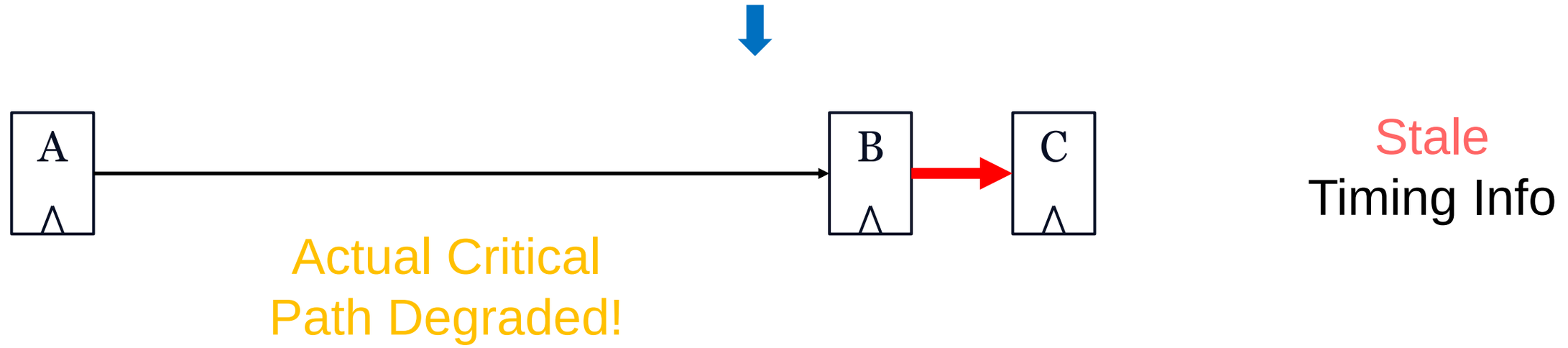
Up-to-date
Timing Info

Placement Critical Path Optimization

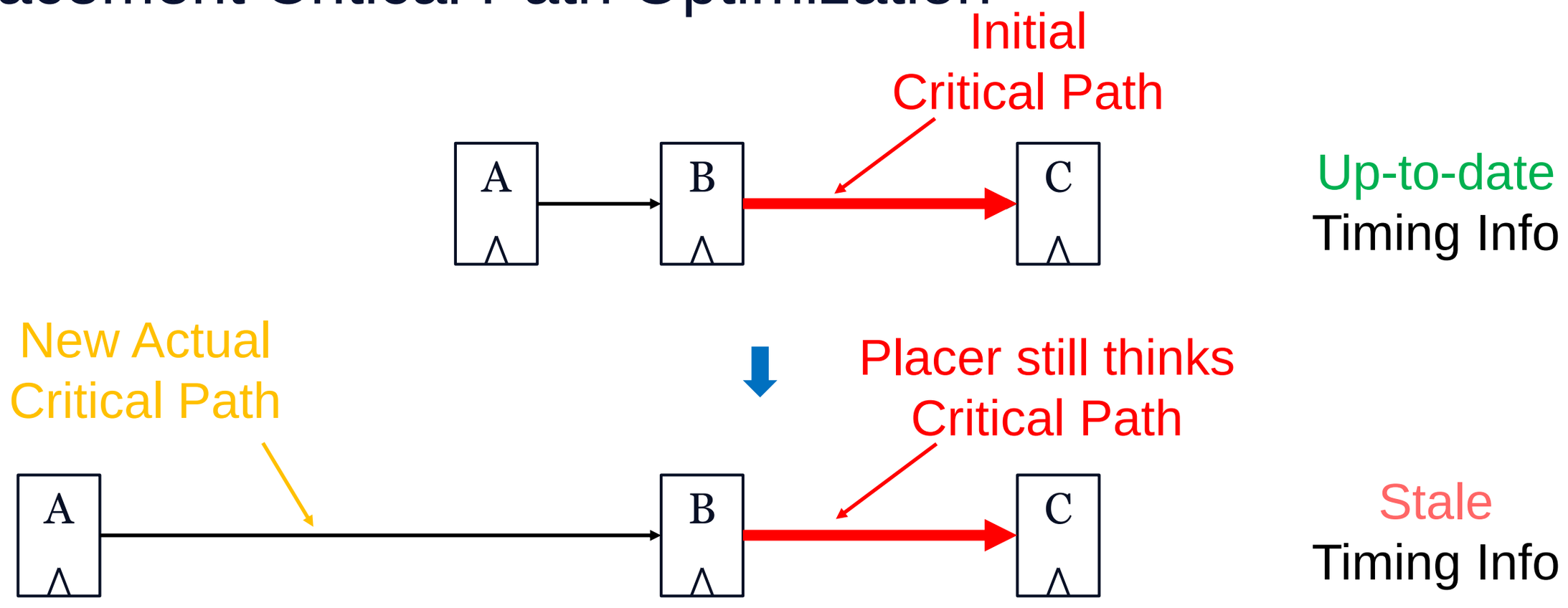


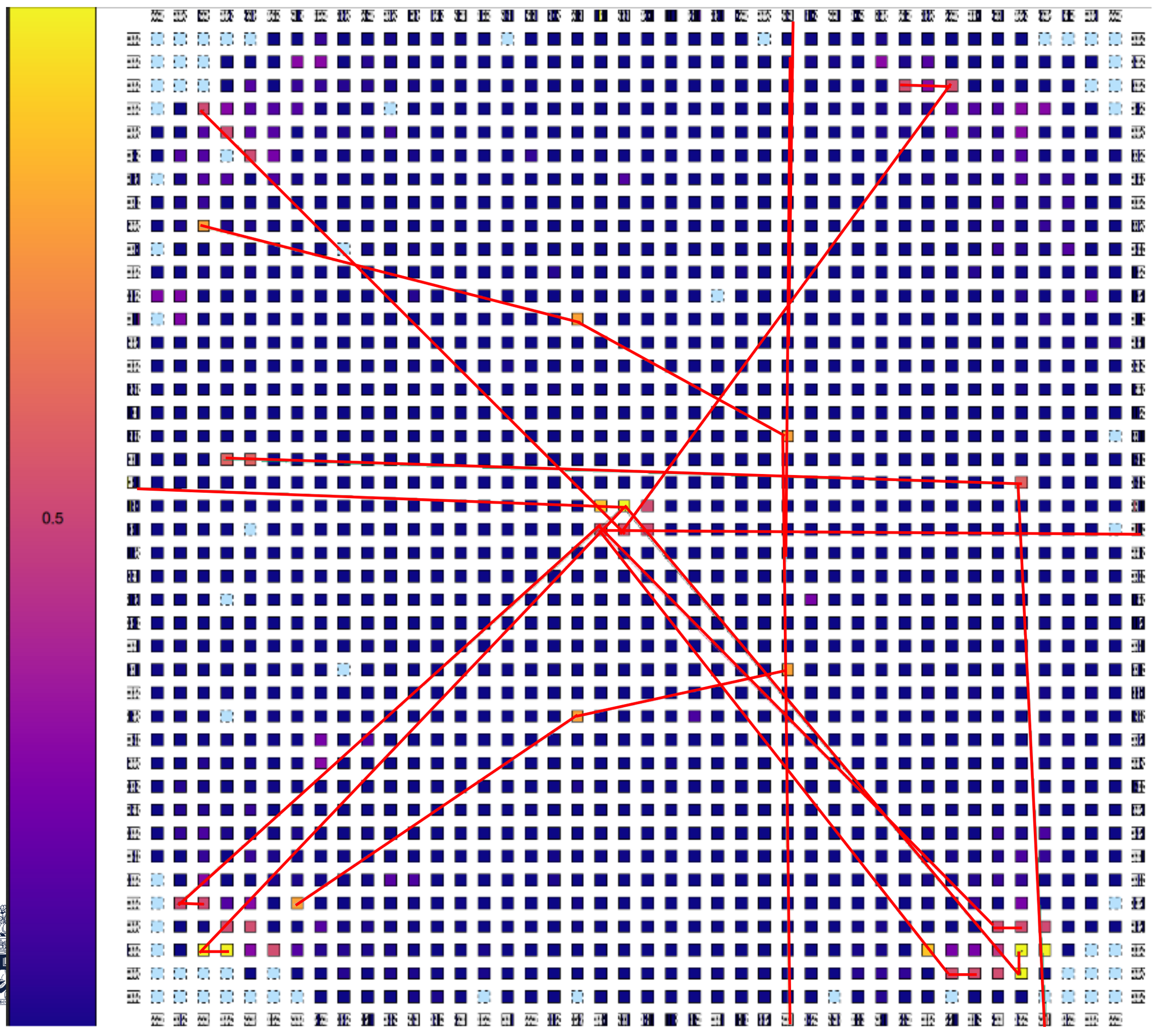
Placement Critical Path Optimization



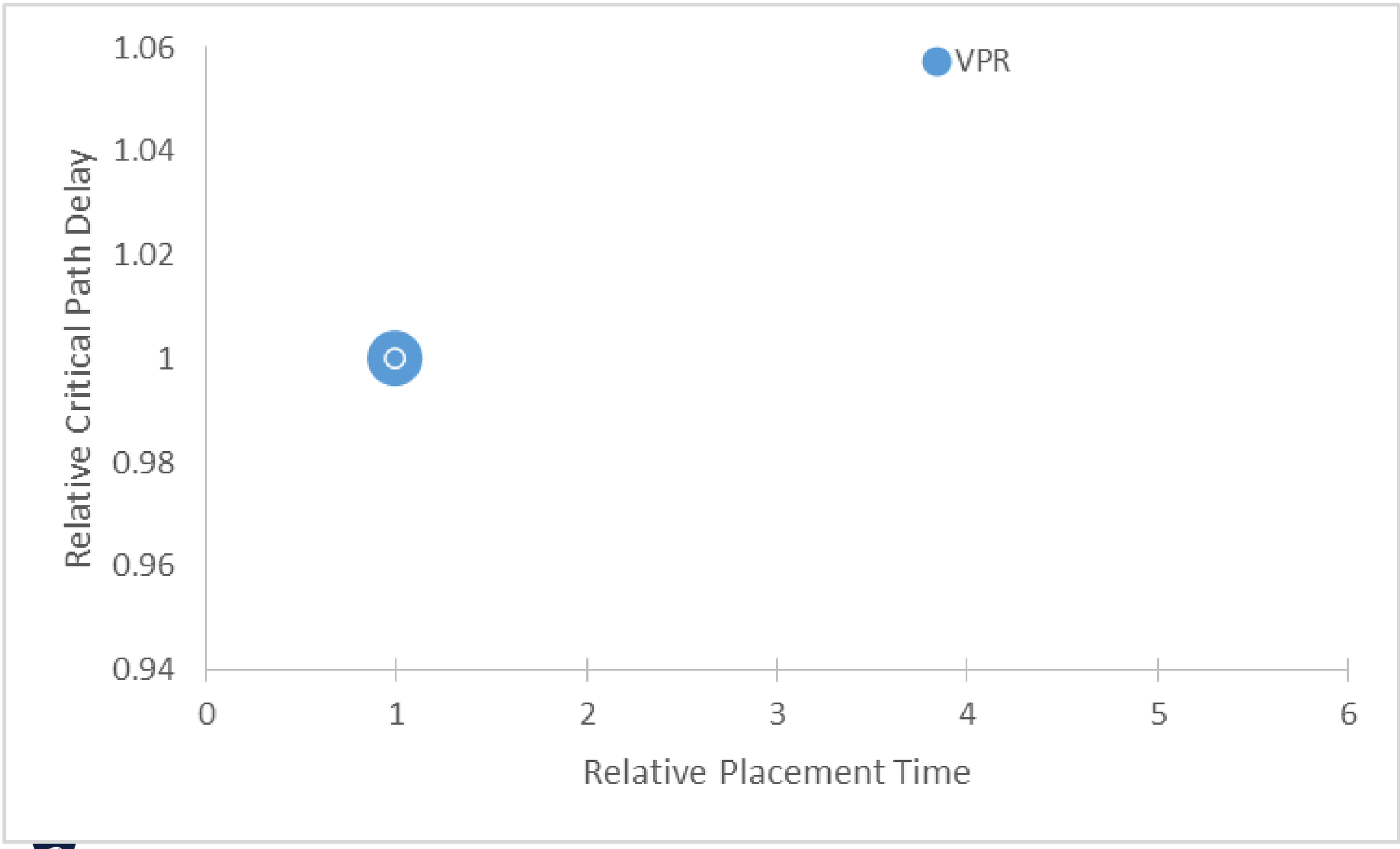


Placement Critical Path Optimization



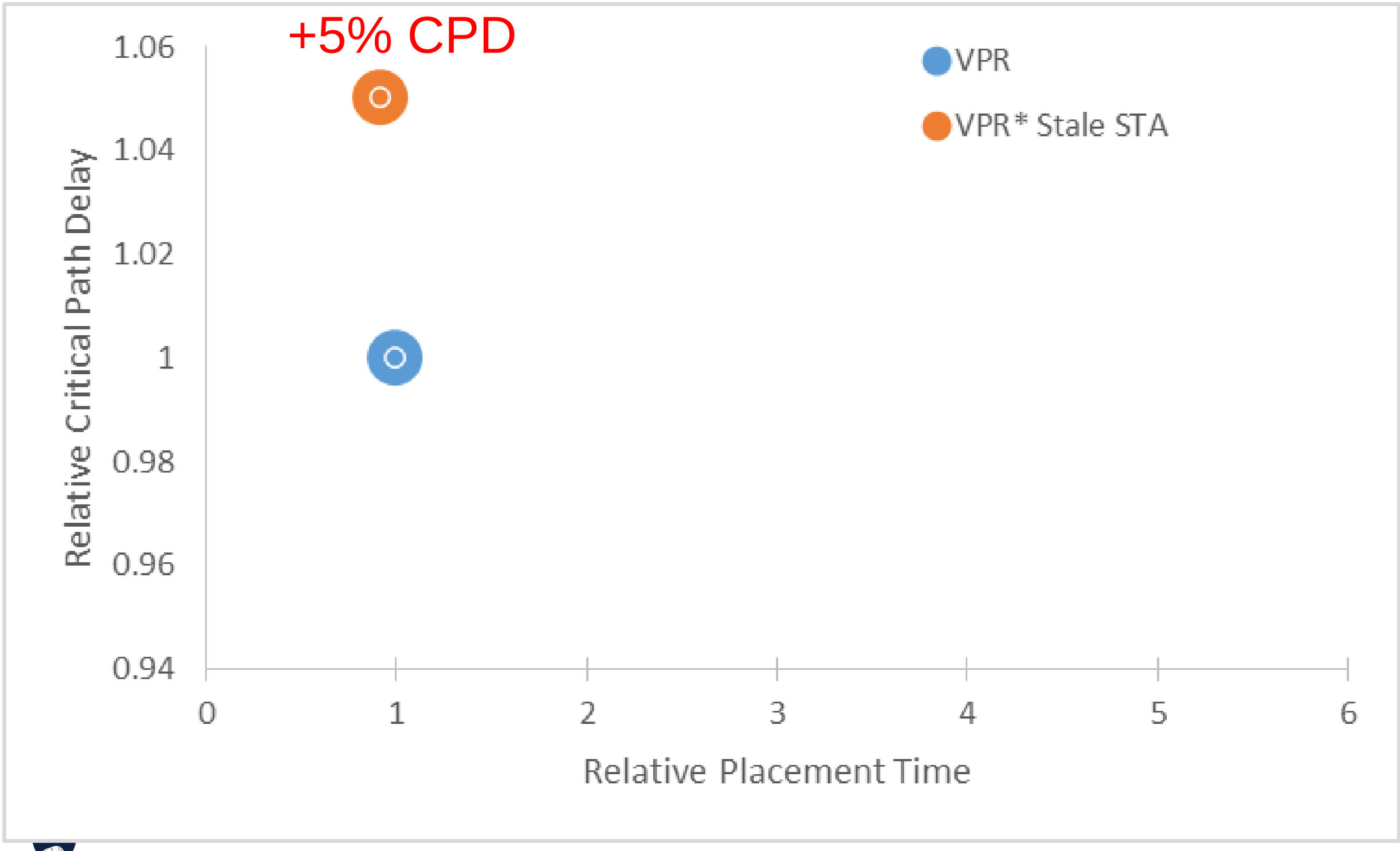

0.5

VPR Placement & STA Frequency



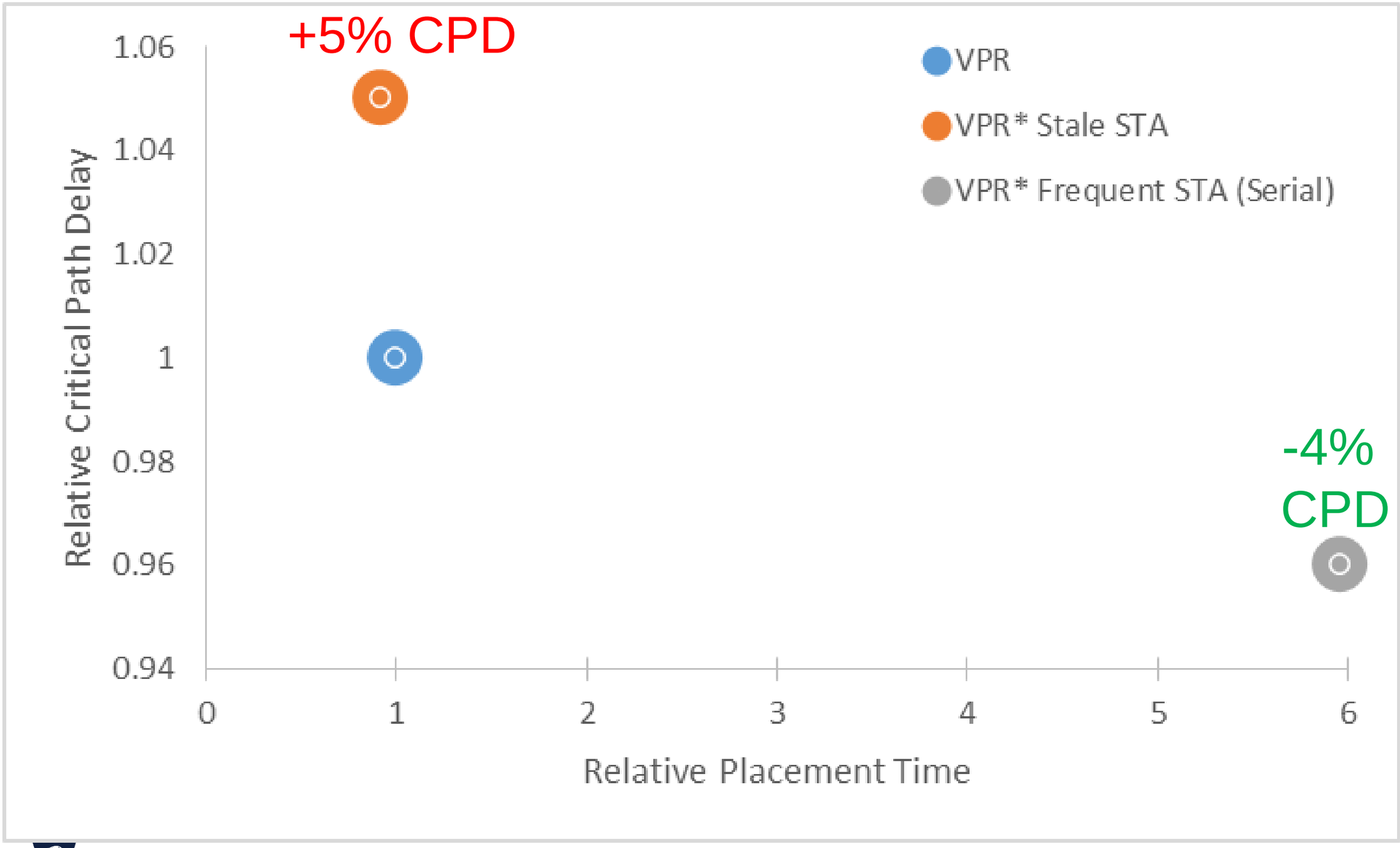
VPR Placement & STA Frequency

- Modify VPR to focus on Critical Paths during late placement



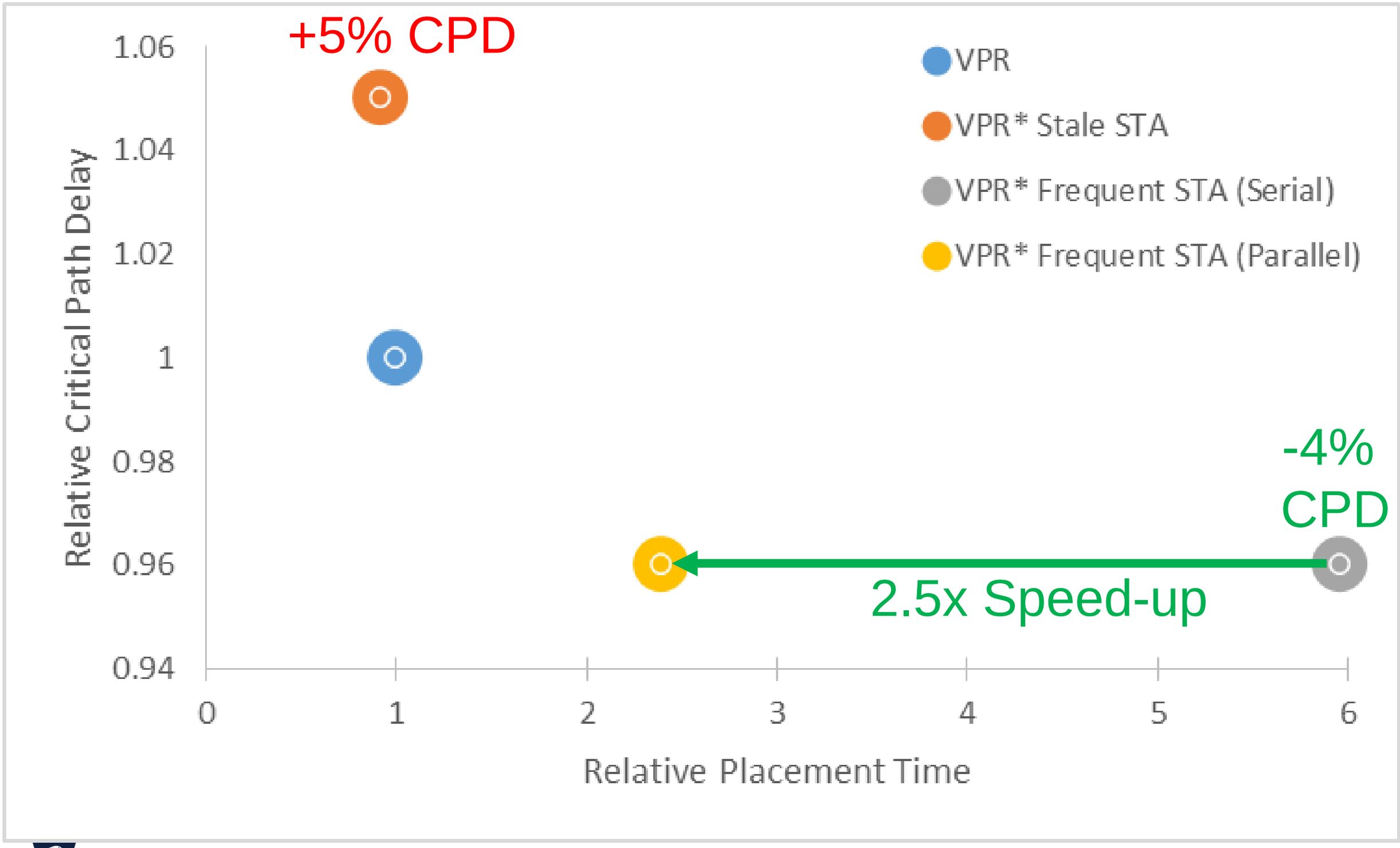
VPR Placement & STA Frequency

- Modify VPR to focus on Critical Paths during late placement



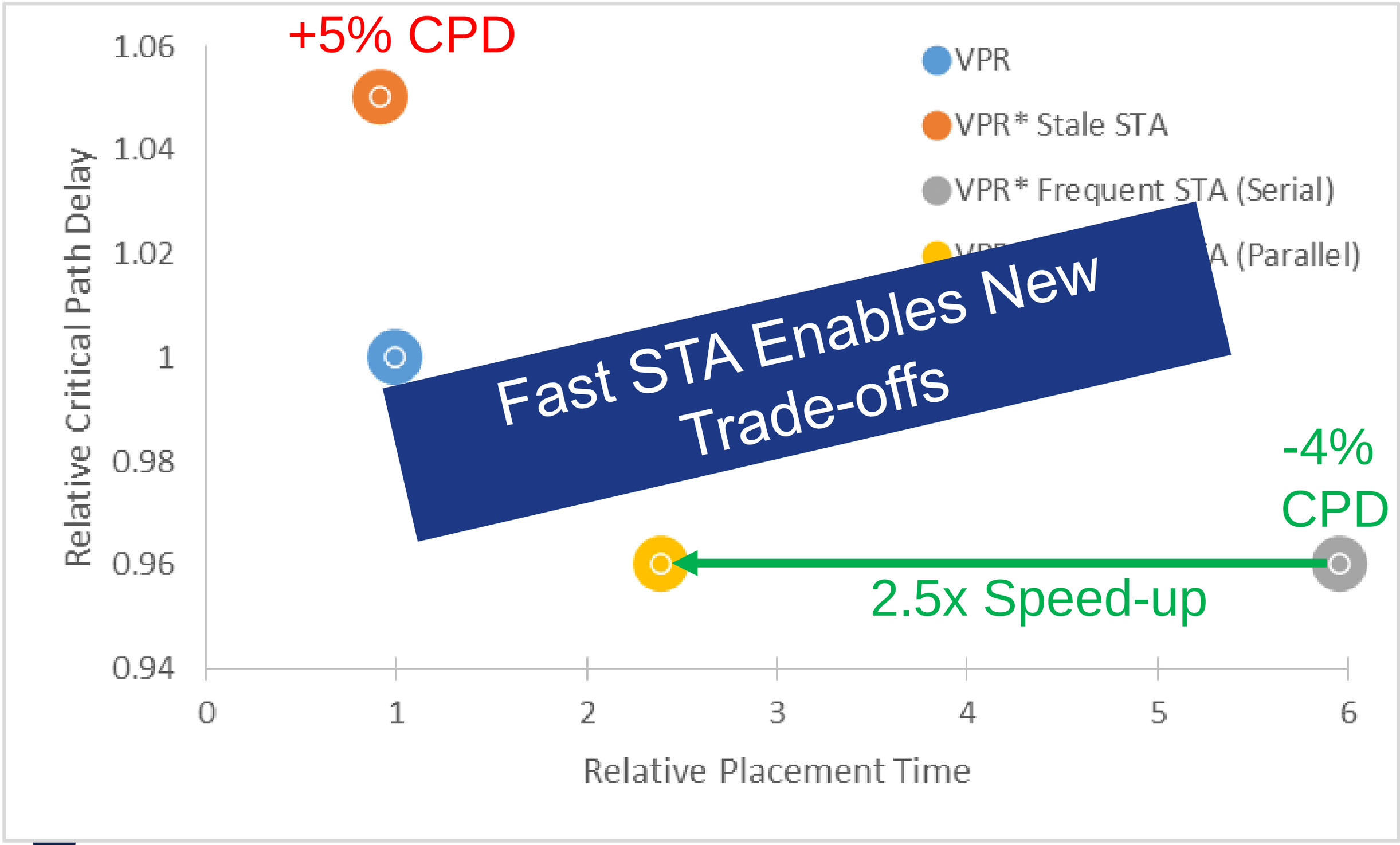
VPR Placement & STA Frequency

- Modify VPR to focus on Critical Paths during late placement



VPR Placement & STA Frequency

- Modify VPR to focus on Critical Paths during late placement



Conclusion

Conclusion

- Tatum STA Engine
  - Full featured & high performance
  - **15-32x** faster than VPR 7 STA with 32 cores
  - Already used by:
    - *VTR 8*
    - *CGRA-ME*
- Fast STA enables new possibilities for timing-driven CAD
  - 4% F_{max} in late placement
  - Routing, Placement, Clustering, Synthesis, ….

Thanks!

Questions?

Email: kmurray@eecg.utoronto.ca

Tatum Open Source Release:

uoft.me/tatum

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

Backup

Multi-corner STA