ECE 669

Parallel Computer Architecture

Lecture 20

Evaluation and Message Passing

UMASS

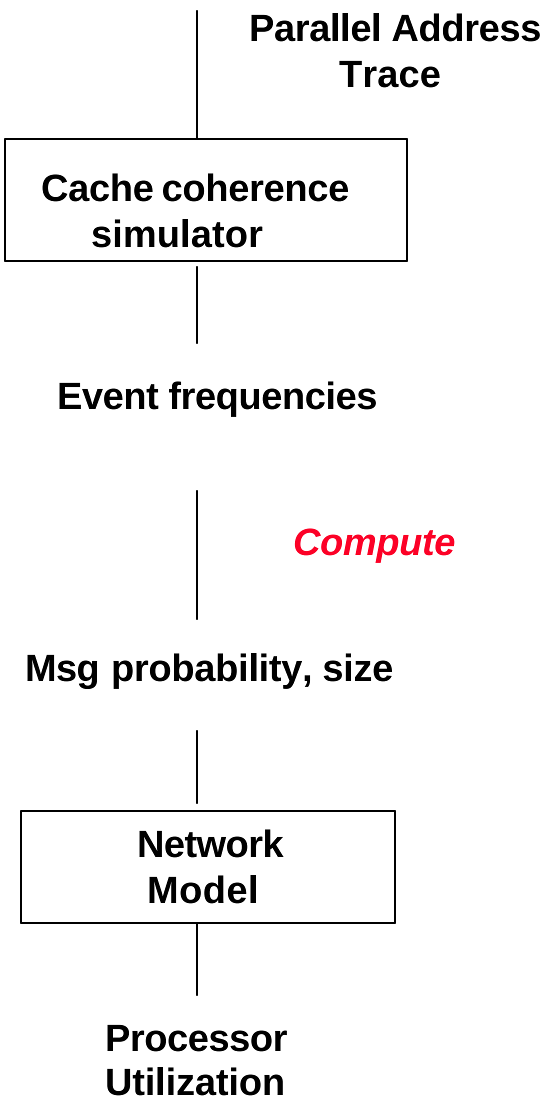
Performance Evaluation

- **Why?**
  - Evaluate tradeoffs
  - Estimate machine performance
  - Measure application behavior
- **How?**
  - Ask an expert --- hire a consultant?
  - Measure existing machines --- (What?!)
  - Build simulators
  - Analytical models
  - Hybrids --- combination of 2-3-4

In the lab

Parallel Traces

Network Model



Example

1. Trace simulation, awk filter

Event	%Prob	out-msg	out-siz	in-msg	in-siz
⋮					
read miss in write mode	1.2507	yes	flits 4	yes	flits * (see formula)
⋮		⋮	⋮	⋮	⋮
other msgs	2.8483	yes	4	yes	4

* : in-siz: 4+ (block size/flit size)

2. Compute network parameters m,B

$$m = \sum_{events} (\text{msgs. per event}) * \frac{(\%\ prob)}{100}$$

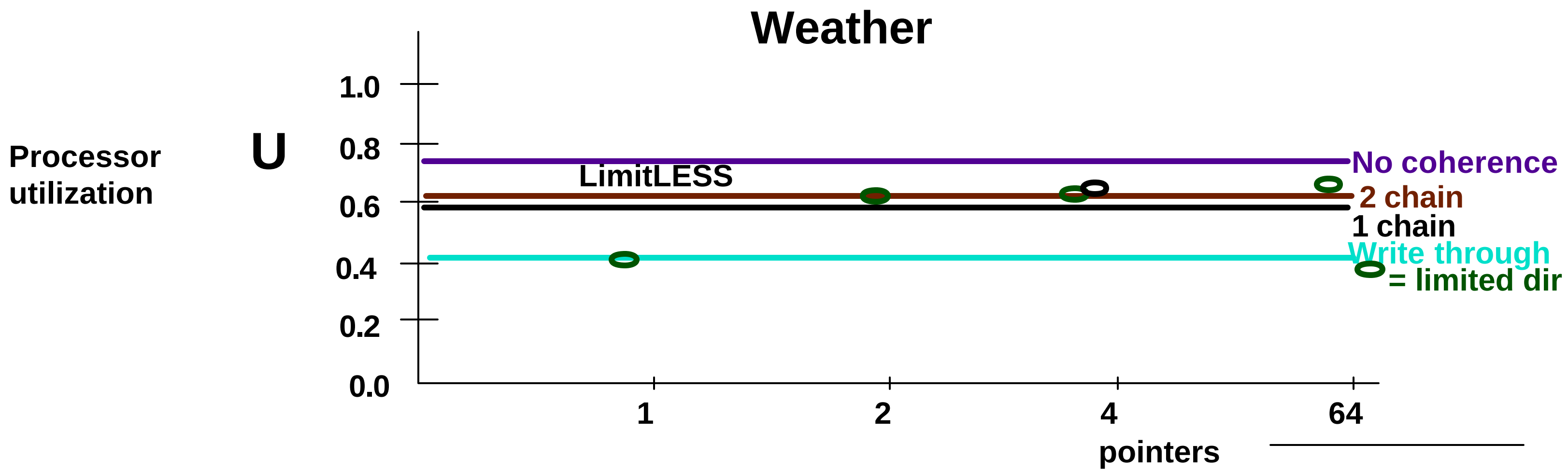
$$B = \frac{\sum_{events} \left[\sum_{msgs} \text{msgsize} \right] * \frac{(\%\ prob)}{100}}{m}$$

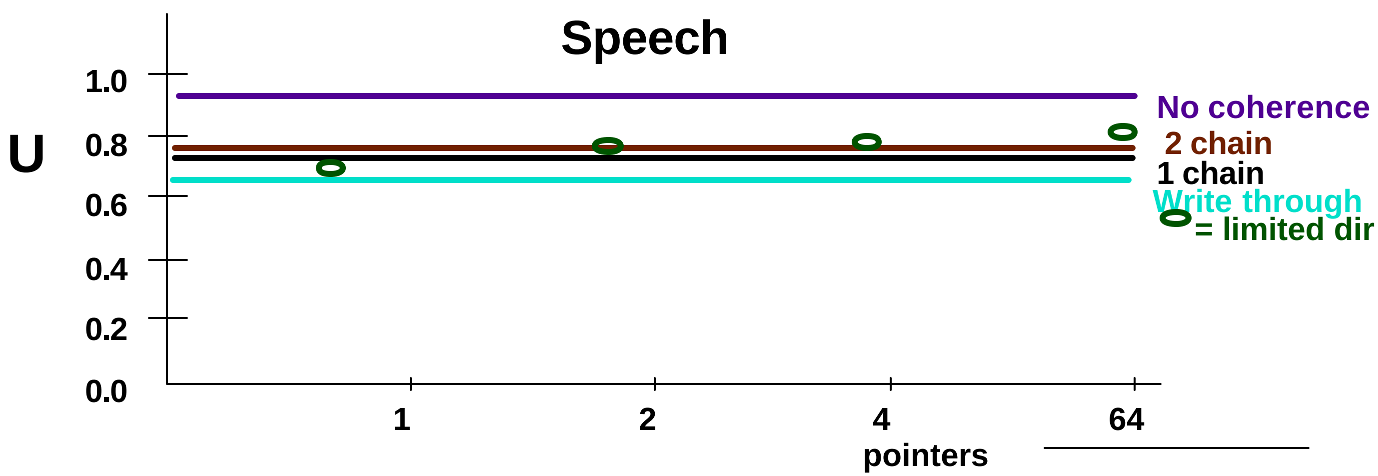
3. Compute processor utilization

Given m,B **and** k_d, n, N

Derive U**.**

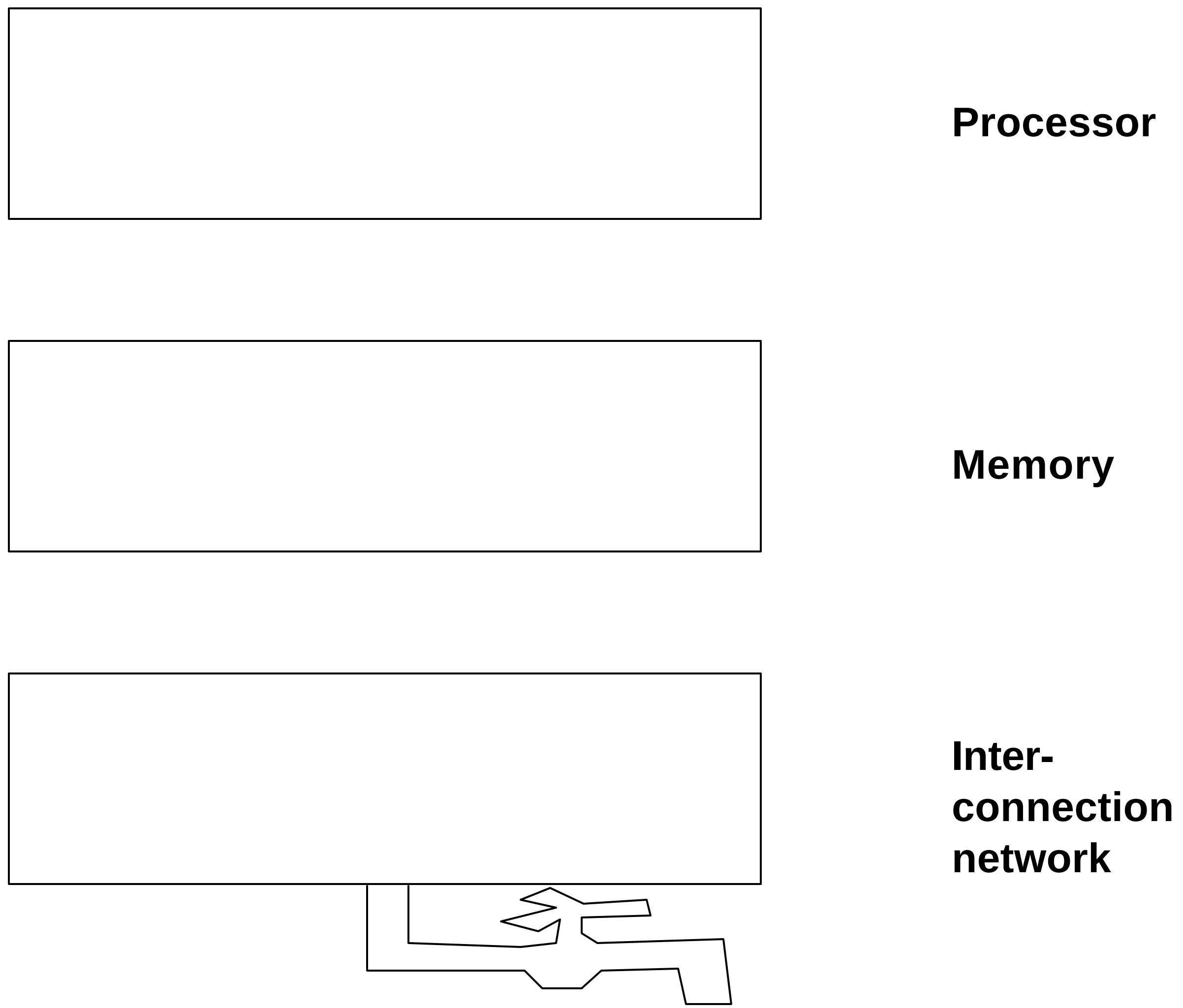
Evaluation





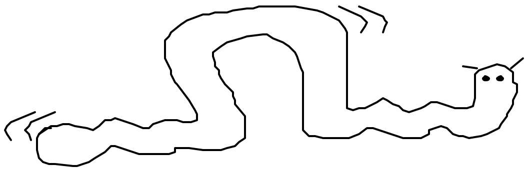
- **Barriers implemented using distributed trees**
- **Read-only sharing marked**

Evaluation



Full system simulation (coupled)

4000x slowdown per node
Very accurate



Compiled application

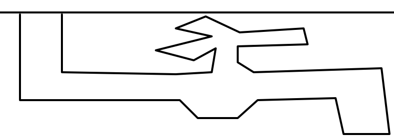
Processor simulator — Processor

Wait, traps | Memory requests | Synchronization requests

Cache and memory systems simulator — Memory

Acknowledgements, responses | Network requests

Interconnection network simulator — Inter-connection network



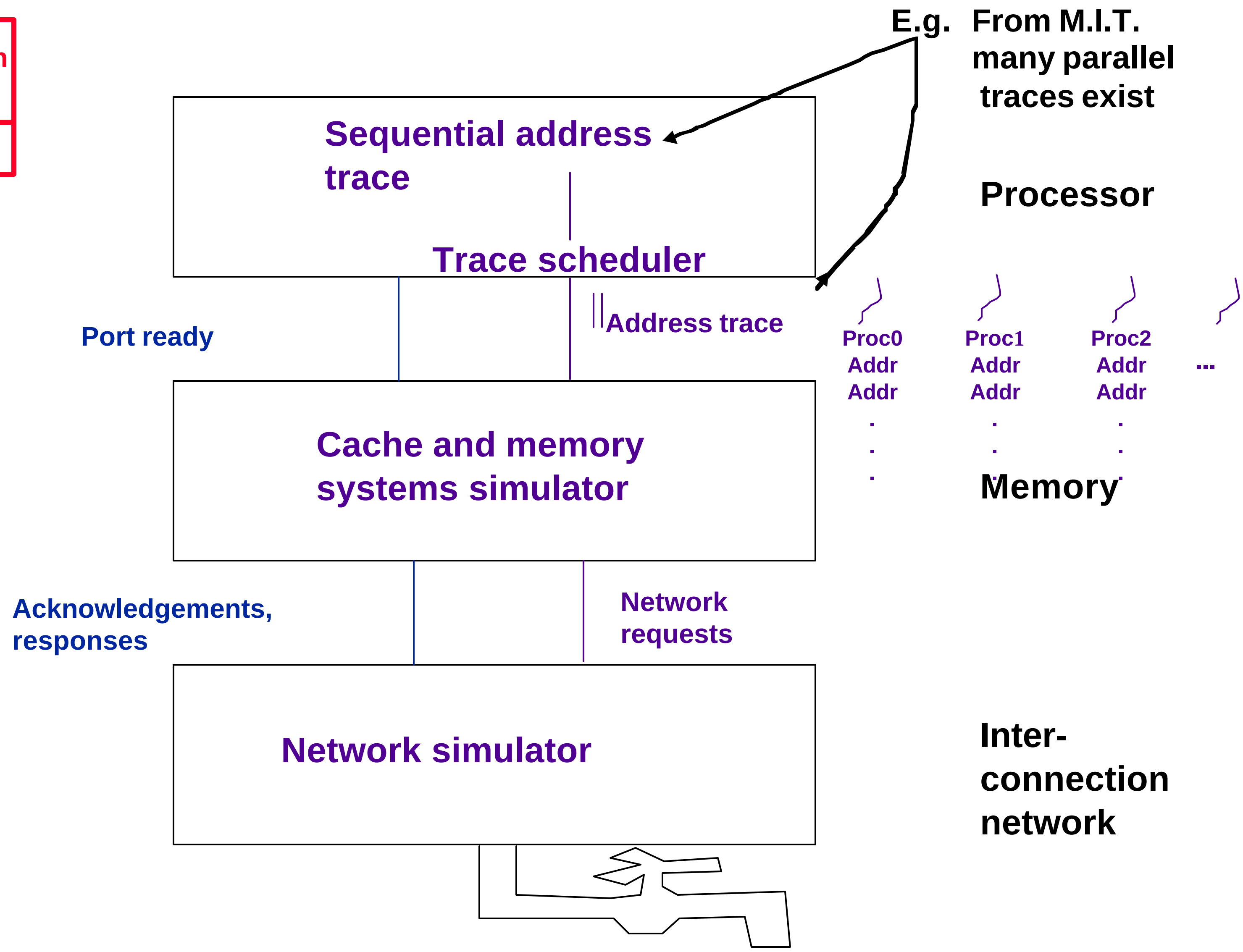
Measures:

- Speedup, runtime
- proc. util.
- net latency
- cache miss rate
- .
- .
- .

Trace-driven simulation (coupled)

2000x slowdown per node

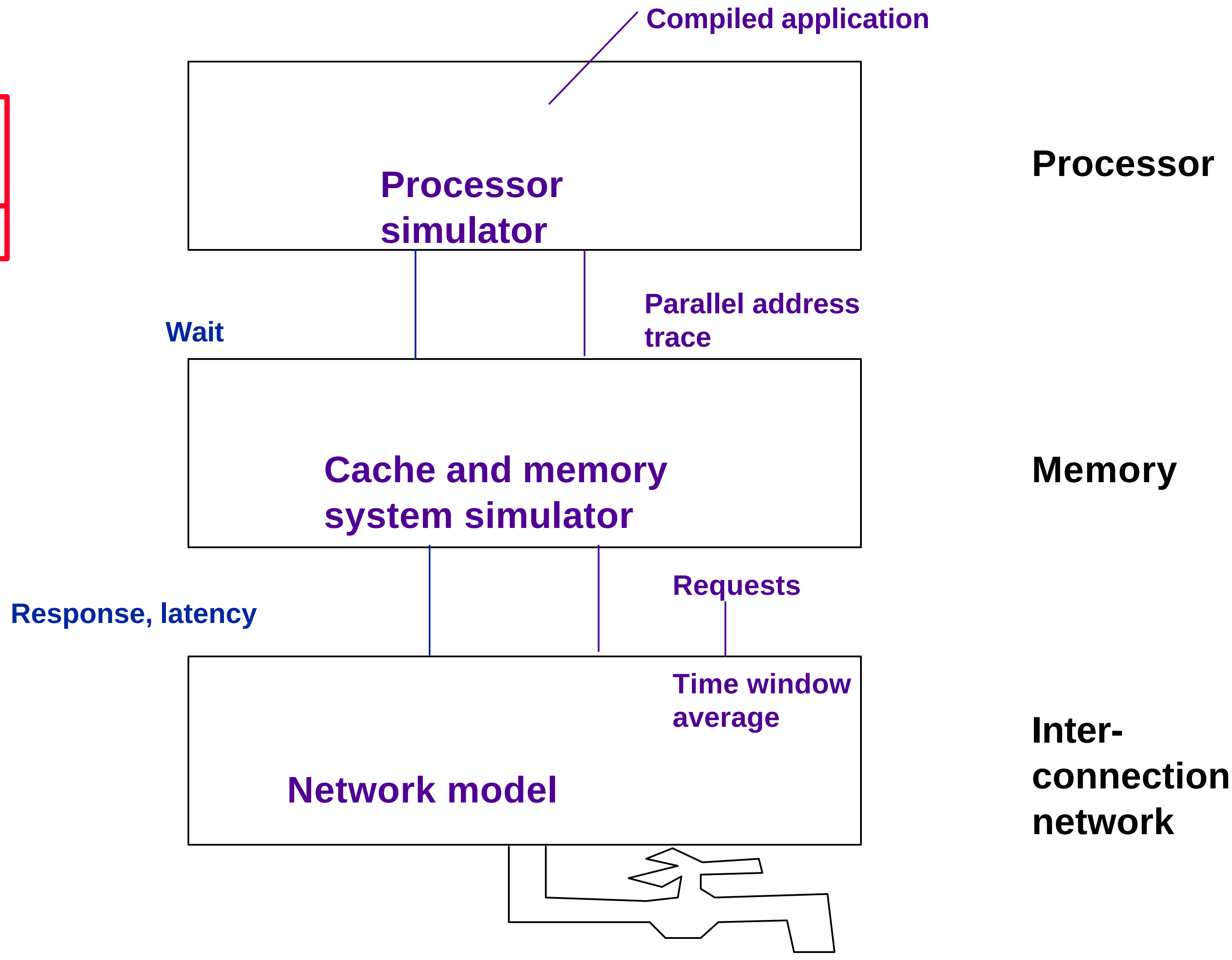
Very accurate



Hybrid Network Model

800x slowdown per node

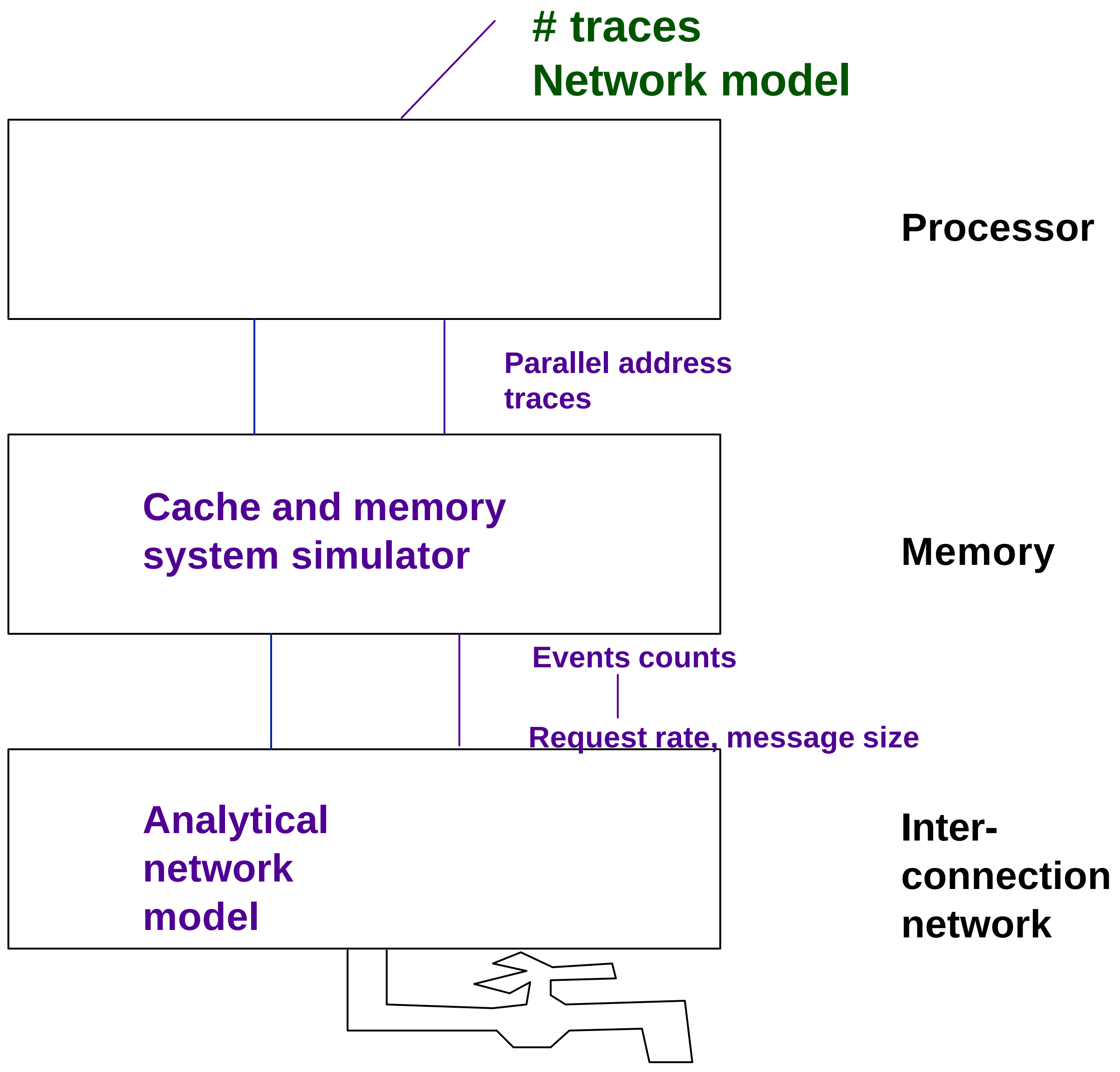
Fair accuracy



Many hybrids possible

200x slowdown per node

Fair accuracy



Hybrid - Network model

100x slowdown
(30x with threads)

Fair accuracy

Compiled application

Direct execution
- Round robin/process
- Switch on mem. req.

Processor

Switch

Memory requests

Cache and memory system simulator

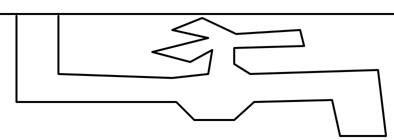
Memory

Responses, latency

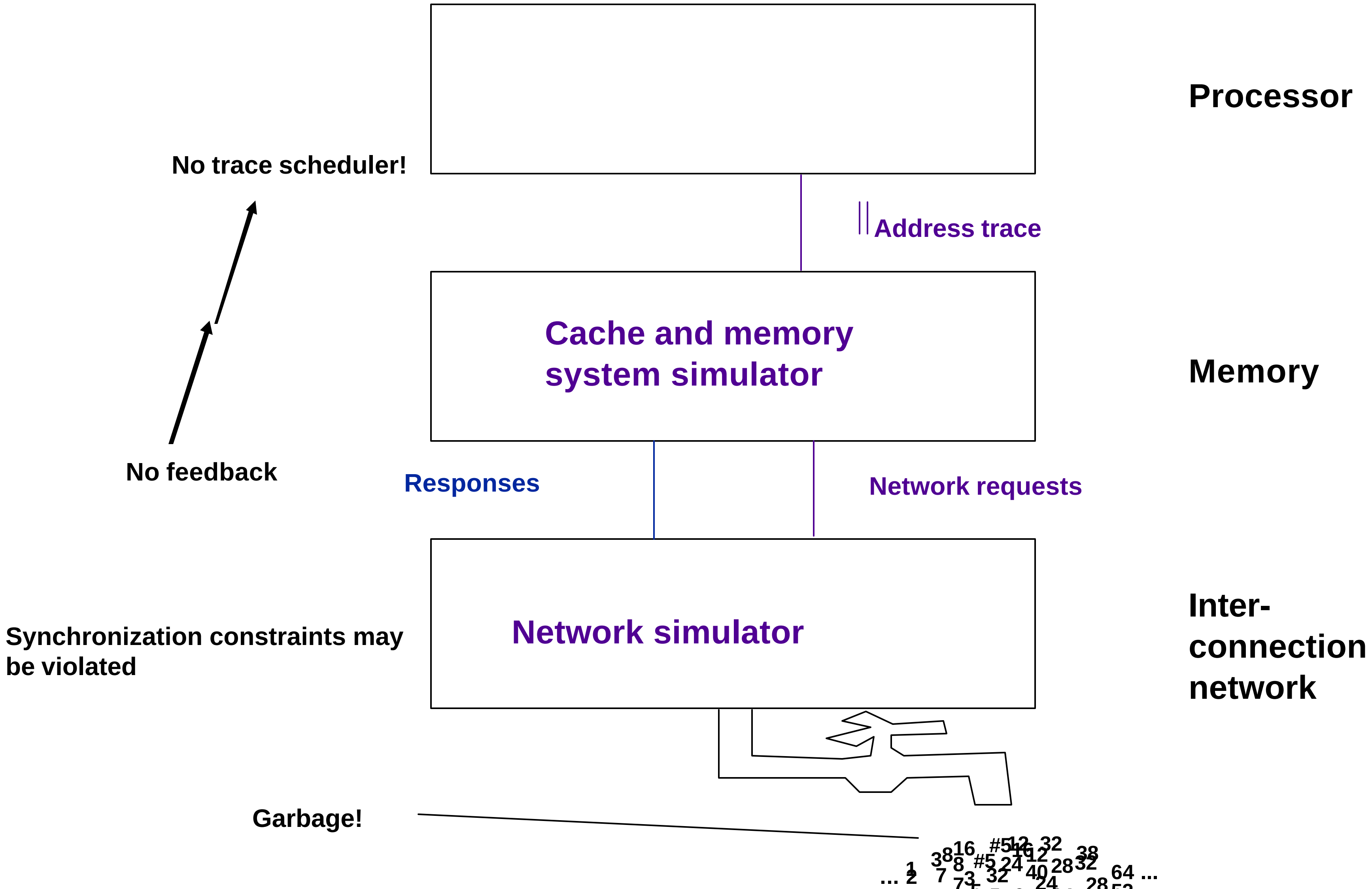
Network requests, time window ave

Network model

Inter-connection network



Trace-driven (decoupled)

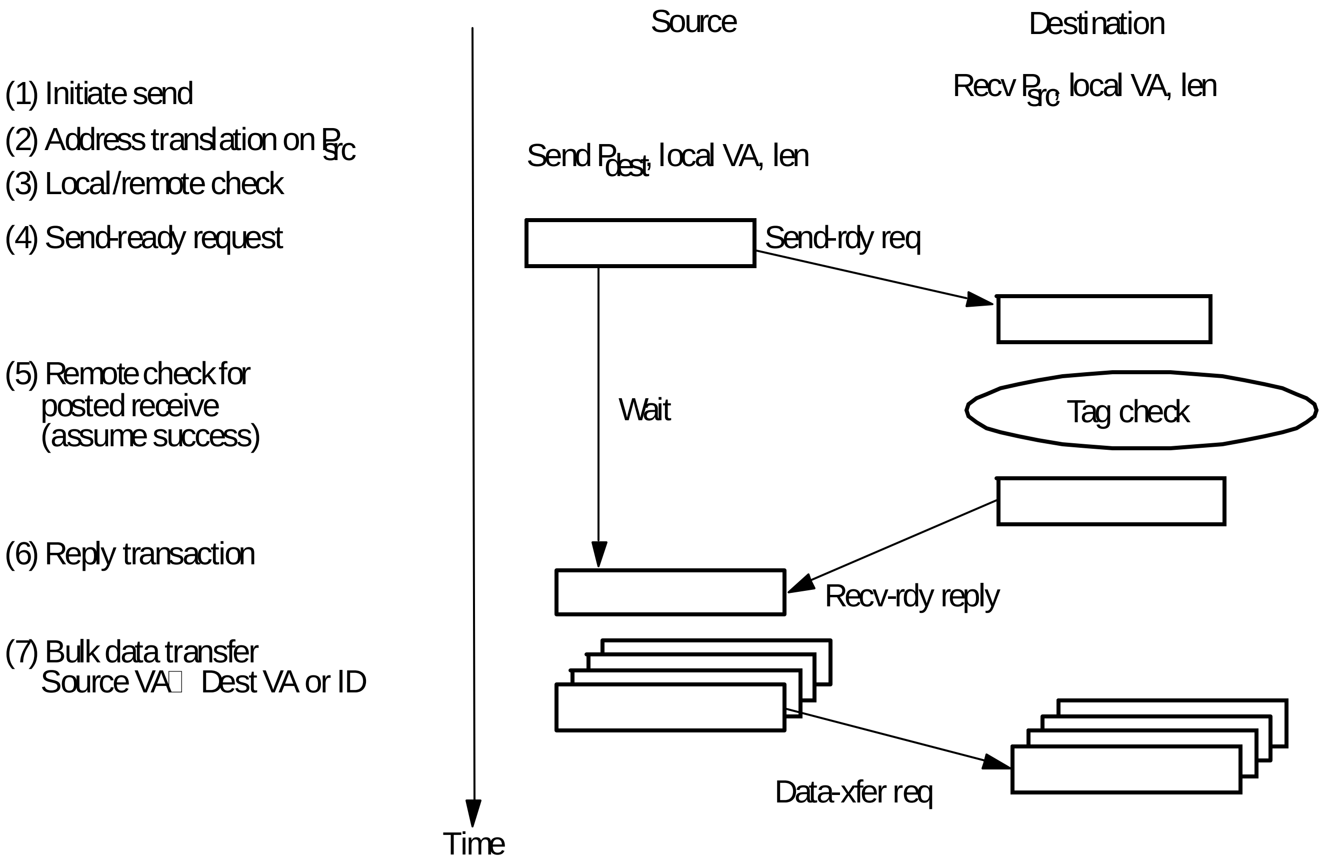


Message passing

- Bulk transfers
- Complex synchronization semantics
  - more complex protocols
  - More complex action

- Synchronous
  - Send completes after matching recv and source data sent
  - Receive completes after data transfer complete from matching send
- Asynchronous
  - Send completes after send buffer may be reused

Synchronous Message Passing

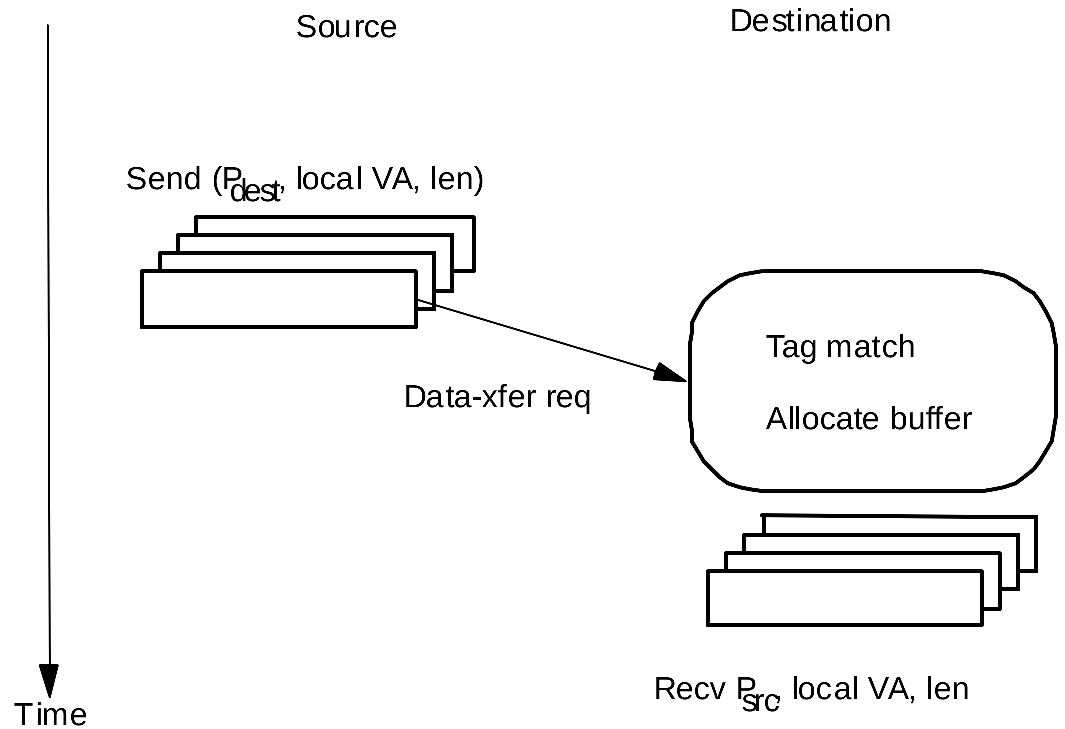


Processor Action?

- **Constrained programming model.**
- **Deterministic!   What happens when threads added?**
- **Destination contention very limited.**
- **User/System boundary?**

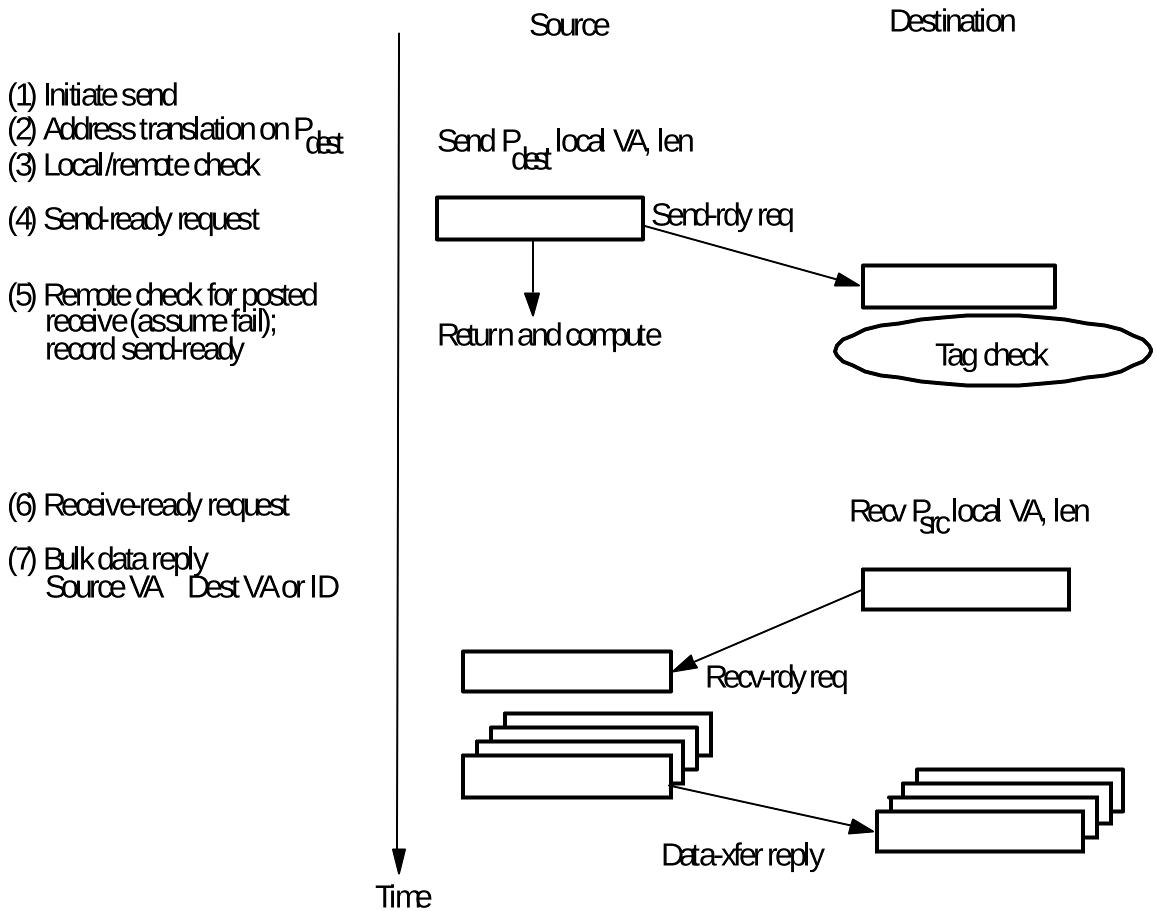
Asynchronous Message Passing: Optimistic

(1) Initiate send
(2) Address translation
(3) Local /remote check
(4) Send data
(5) Remote check for posted receive; on fail, allocate data buffer



- More powerful programming model
- Wildcard receive => non-deterministic
- Storage required within msg layer?

Asynchronous Message Passing: Conservative

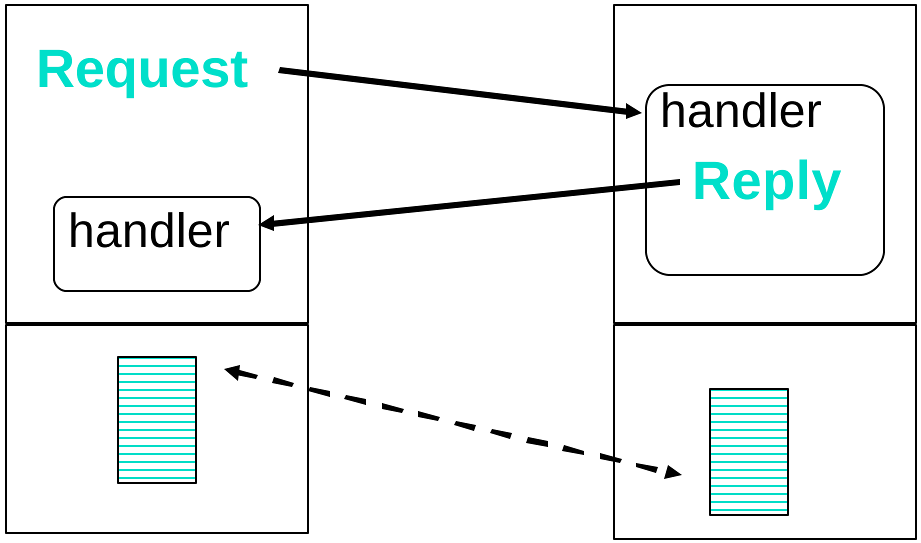


- **Where is the buffering?**
- **Contention control? Receiver initiated protocol?**
- **Short message optimizations**

Key Features of Message Passing Abstraction

- **Source knows send data address, dest. knows receive data address**
  - after handshake they both know both
- **Arbitrary storage “outside the local address spaces”**
  - may post many sends before any receives
  - non-blocking asynchronous sends reduces the requirement to an arbitrary number of descriptors
    - fine print says these are limited too
- **Fundamentally a 3-phase transaction**
  - includes a request / response
  - can use optimisitic 1-phase in limited “Safe” cases
    - credit scheme

Active Messages



- User-level analog of network transaction
  - transfer data packet and invoke handler to extract it from the network and integrate with on-going computation
- Request/Reply
- Event notification: interrupts, polling, events?
- May also perform memory-to-memory transfer

Common Challenges

- **Input buffer overflow**
  - **N-1 queue over-commitment => must slow sources**
  - **reserve space per source (credit)**
    - **when available for reuse?**
      - Ack or Higher level
  - **Refuse input when full**
    - **backpressure in reliable network**
    - **tree saturation**
    - **deadlock free**
    - **what happens to traffic not bound for congested dest?**
  - **Reserve ack back channel**
  - **drop packets**
  - **Utilize higher-level semantics of programming model**

Summary

- **Evaluation – important to understand intermediate messages in cache protocol**
- **Message sizes may vary based on function**
- **Two main types of message passing protocols**
  - **Synchronous and asynchronous**
- **Active messages involve remote operations**
- **Message techniques depend on network reliability**