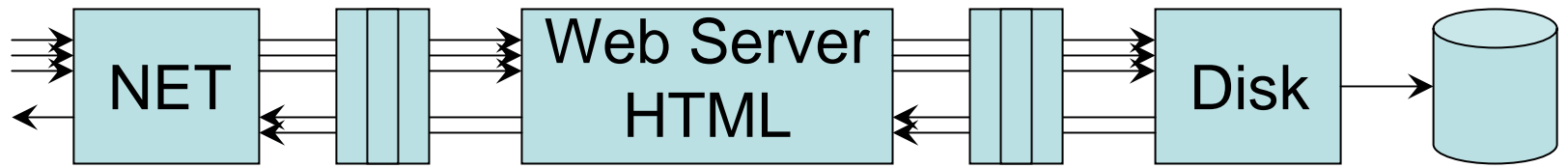


MIT OpenCourseWare
<http://ocw.mit.edu>

6.033 Computer System Engineering
Spring 2009

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.



Sequence Coordination

```
eventcount → integer  
wait(eventcount, value)  
notify(eventcount)
```

```
if ec ≤ value  
    wait
```

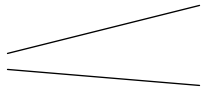
Performance

Concurrency

Caching

Scheduling

Performance Metrics

Capacity: Amt of resource  size in GBS
instrs/sec

Utilization: % of capacity using

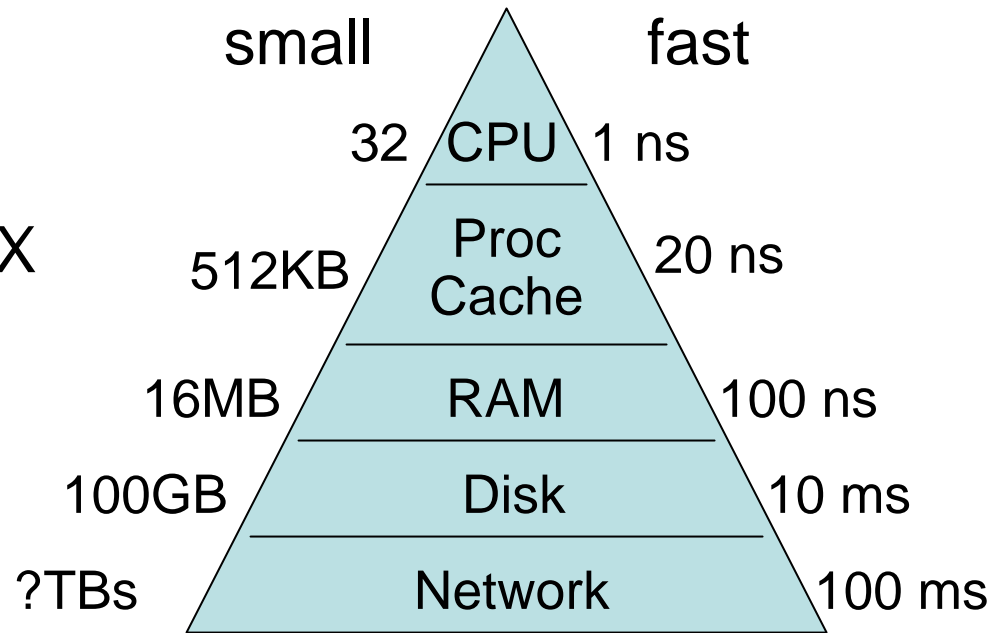
Latency: time for a req. to complete

Throughput: req/sec

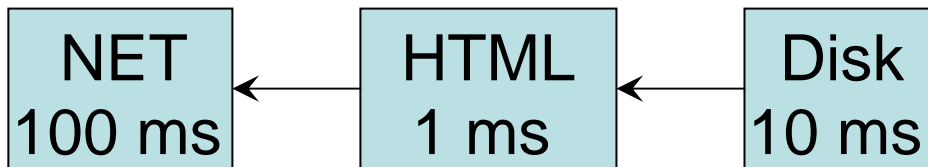
Performance Bottlenecks

I/O Bottleneck

$10^8 = 100 \text{ MX}$

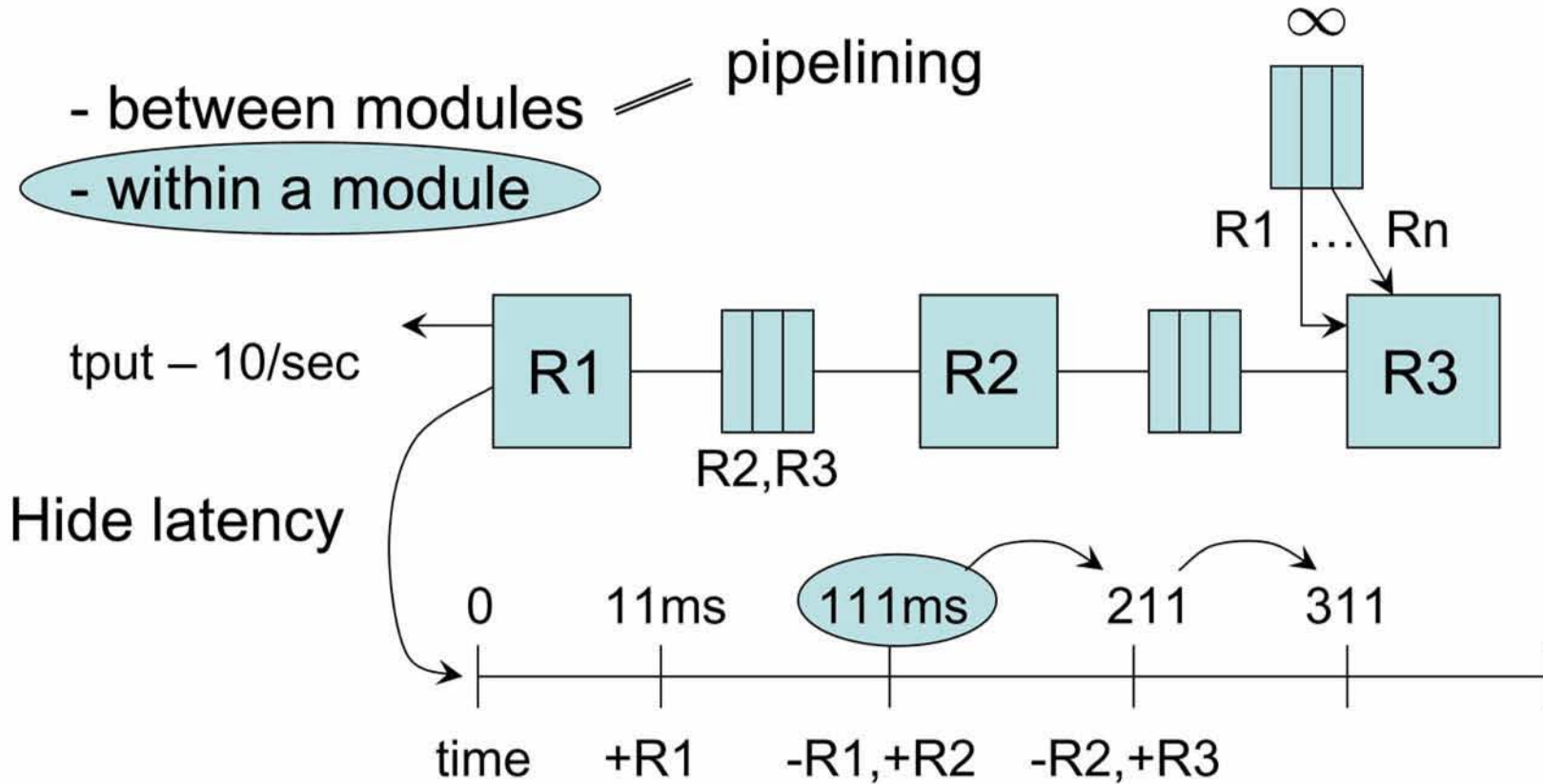


111 ms

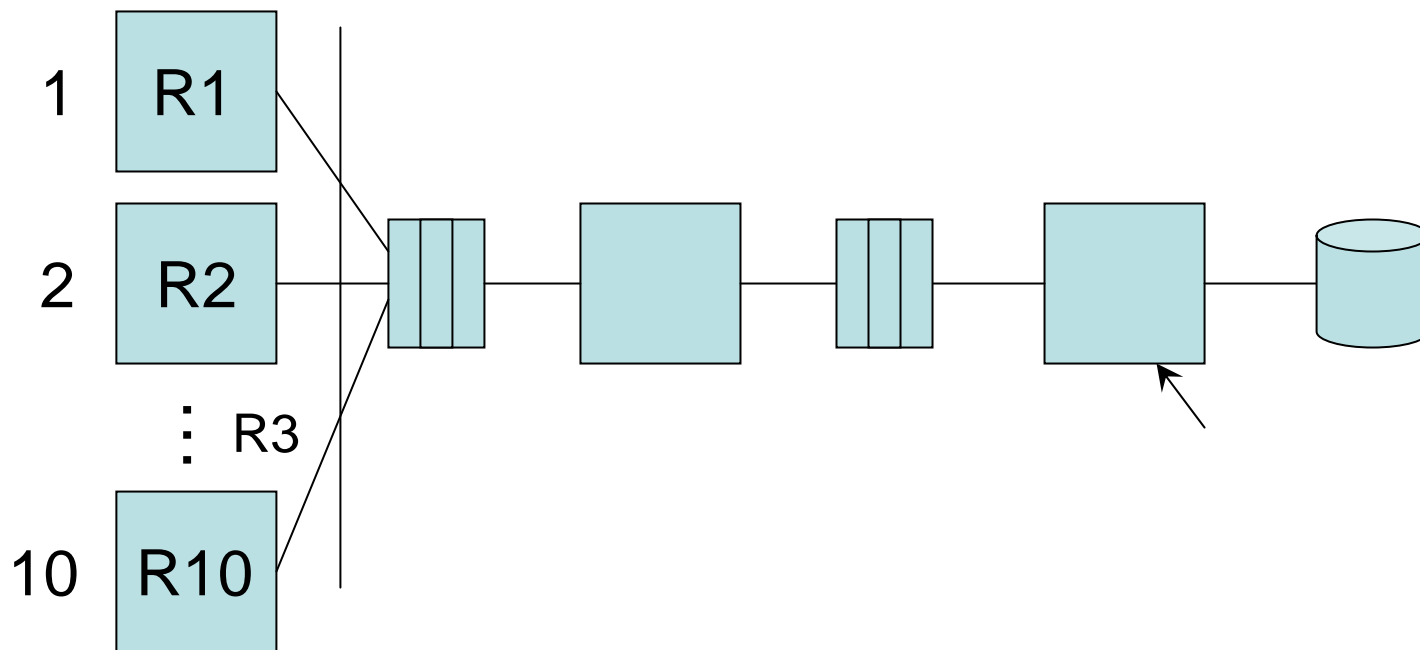
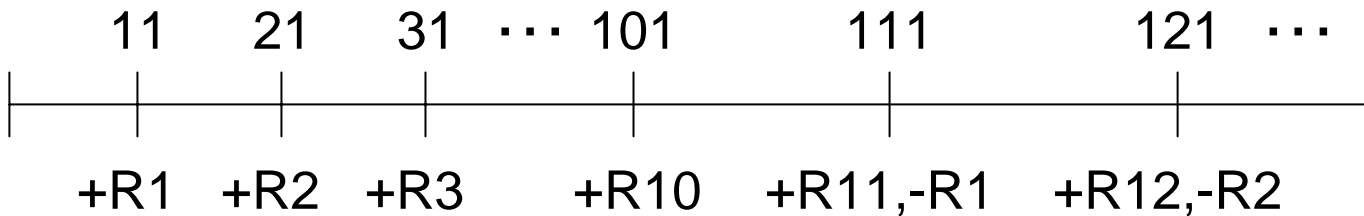


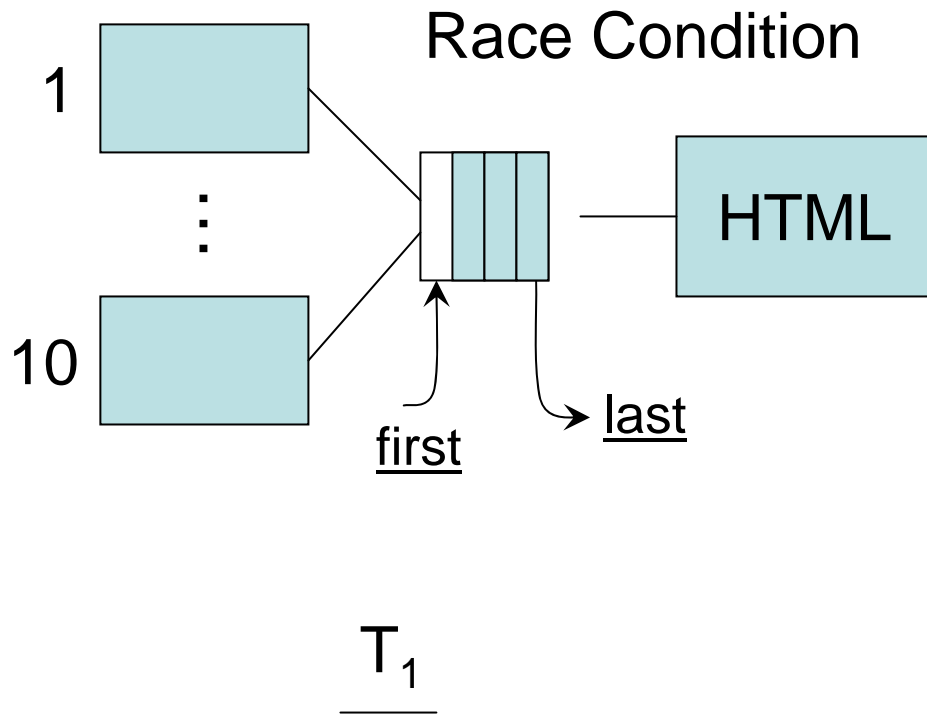
Concurrency

- between modules \equiv pipelining
- within a module



100/sec
tput





Outcomes – 1) OK

```

→page ← buf[first]
→first ← first + 1
→ret. page

```

```

→page ← buf[first]
→first ← first + 1
→return

```

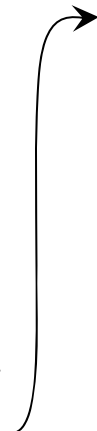
Isolation Primitives

|| atomic
|| isolate → locks
lock → set, unset
ACQUIRE
RELEASE

```
lock t1  
ACQ(t1)  
page ← buf[first]  
first ← first + 1  
REL(t1)
```

RSL – read + set lock

```
held = false  
while(!held)  
    held = RSL(t1)  
end
```



Caching