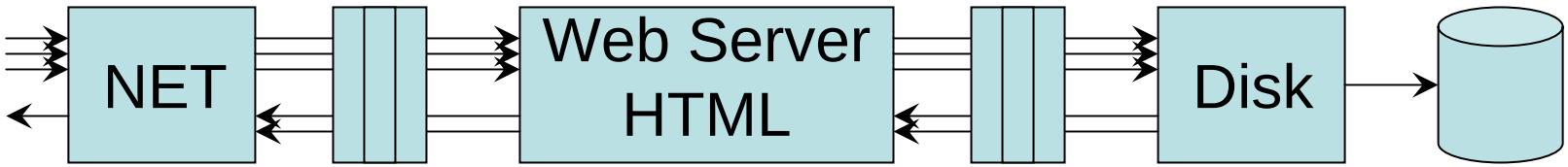

NET
Web Server
HTML
Disk

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

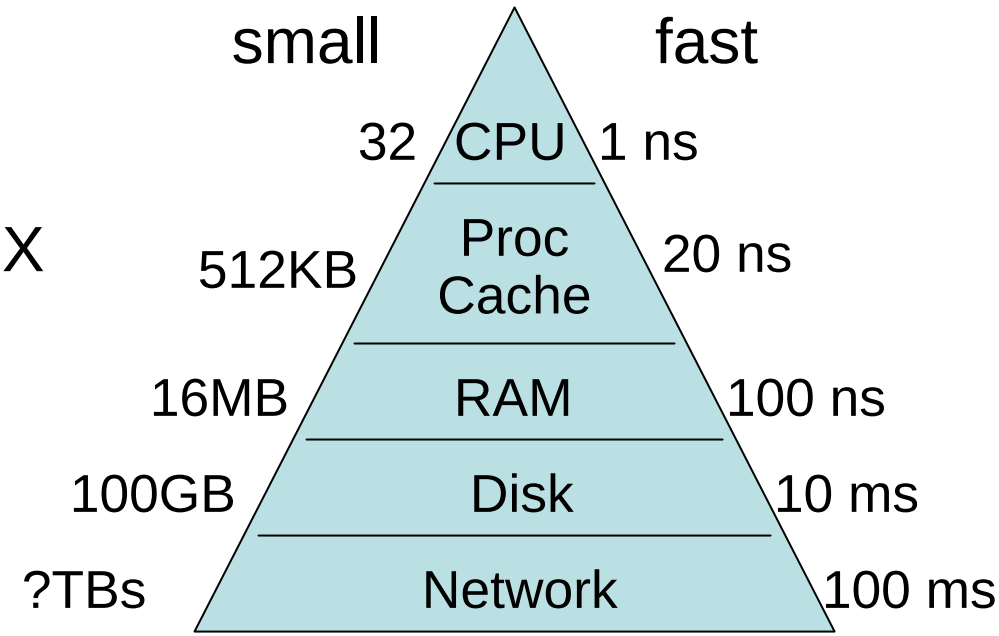
Utilization: % of capacity using

Latency: time for a req. to complete

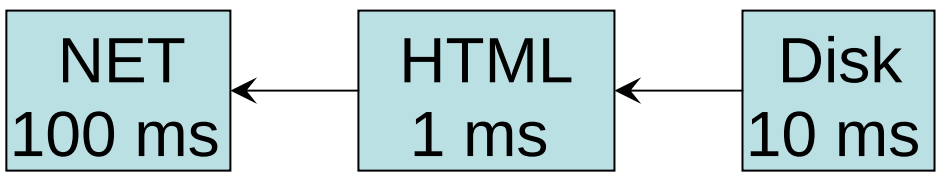
Throughput: req/sec

Performance Bottlenecks
I/O Bottleneck

10^8 = 100 MX



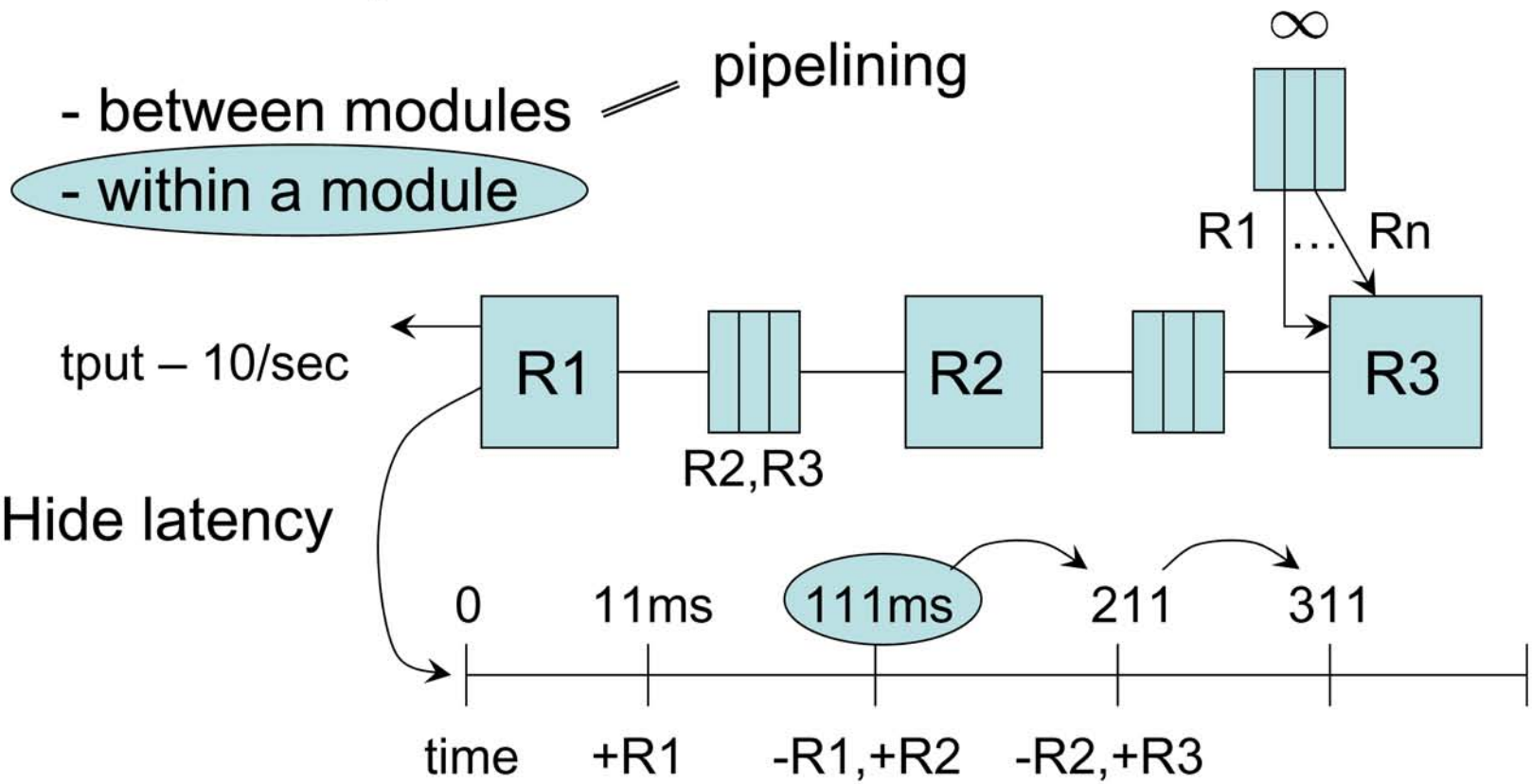
111 ms

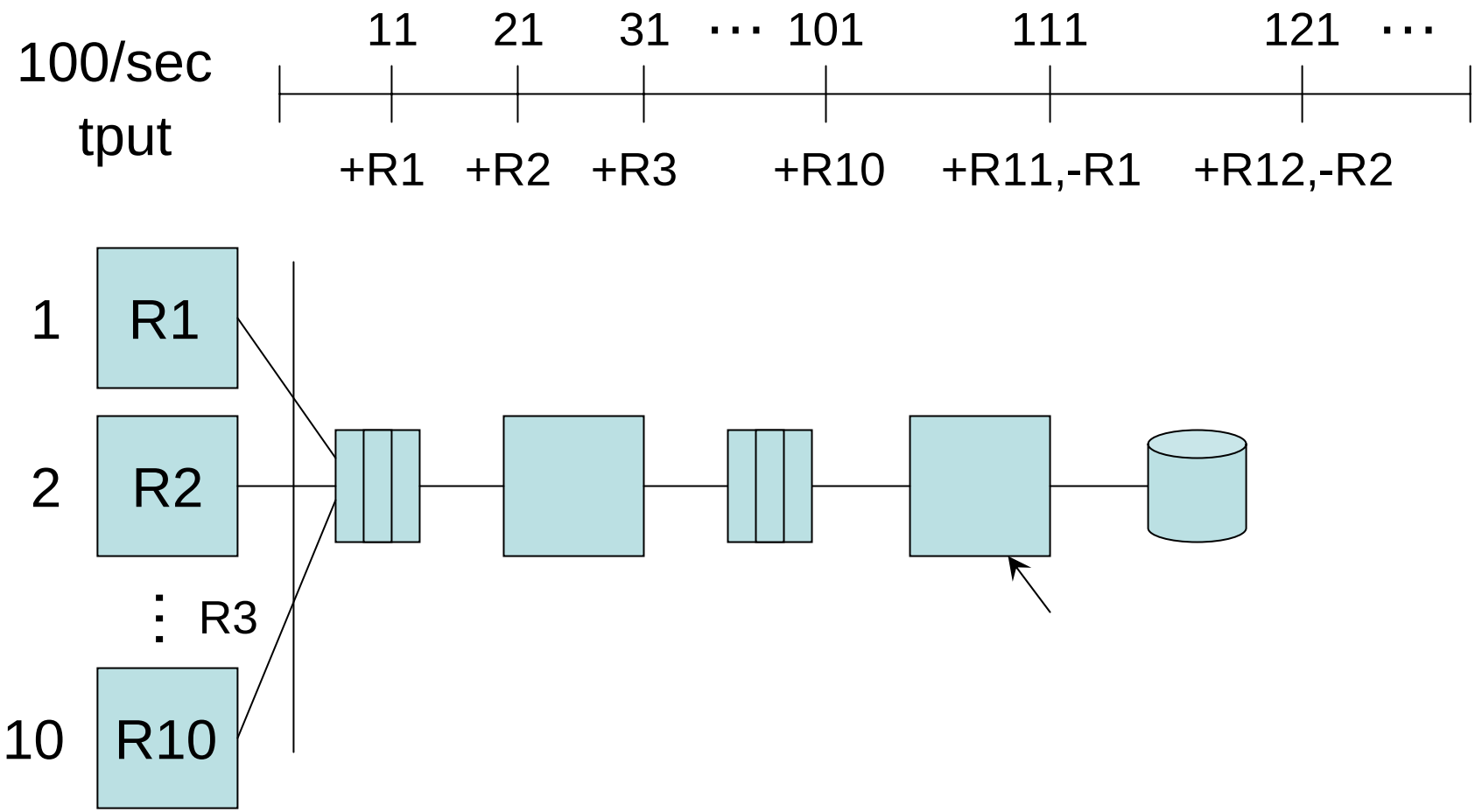


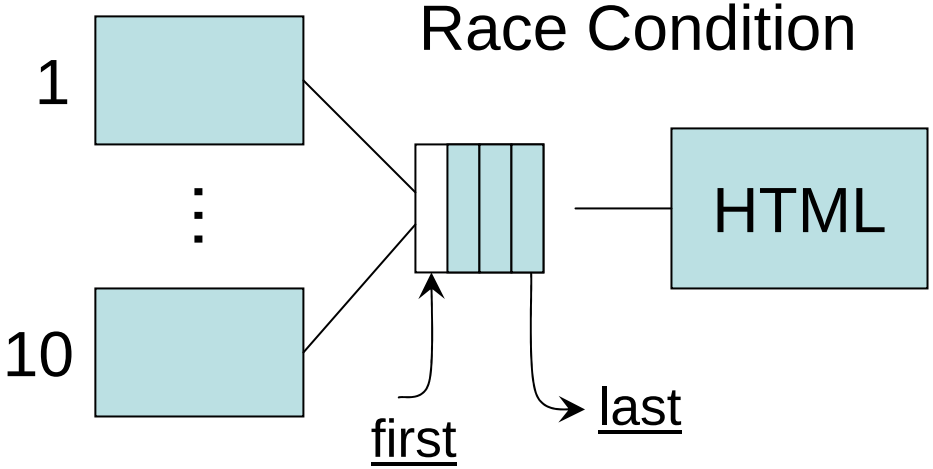
Concurrency

- between modules
- within a module

pipelining




100/sec
tput
11
21
31
101
111
121
+R1
+R2
+R3
+R10
+R11,-R1
+R12,-R2
1
R1
2
R2
R3
10
R10



T_1

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

T_2

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

Isolation Primitives

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

RSL – read + set lock

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

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

Caching