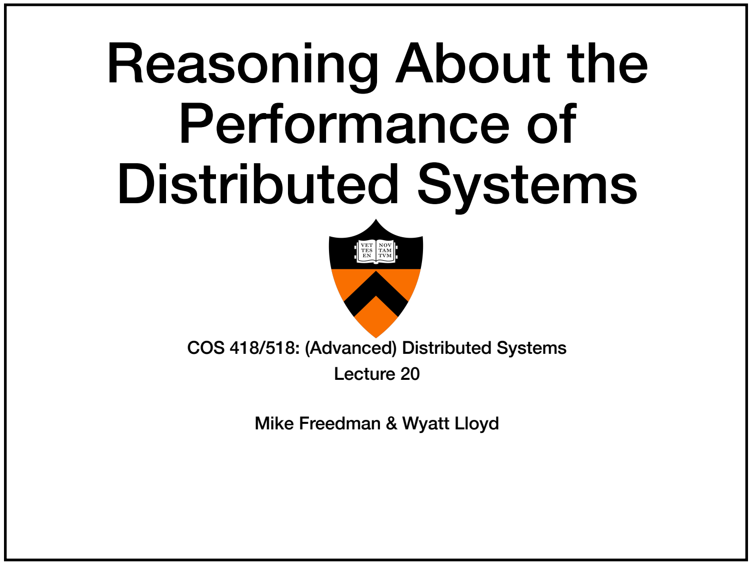
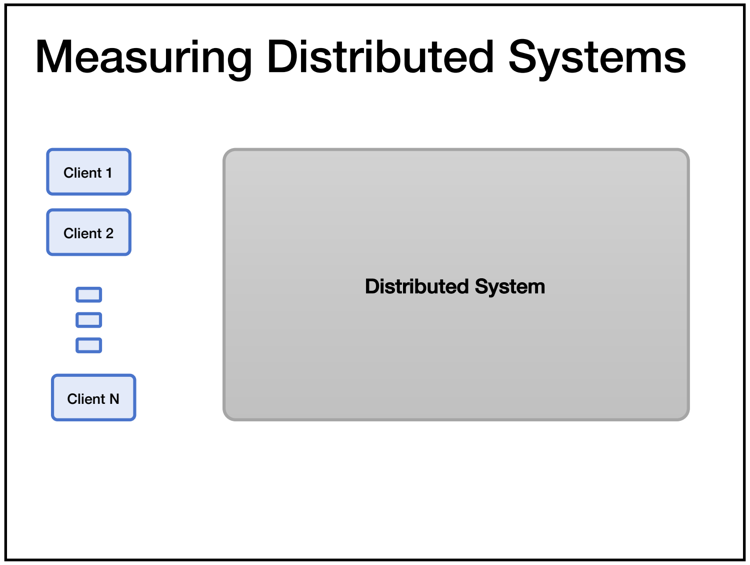


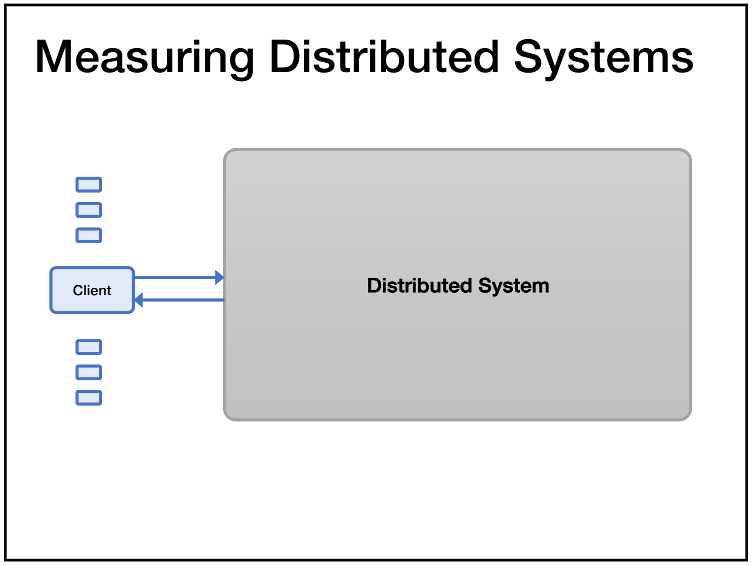
Reasoning About the Performance of Distributed Systems

COS 418/518: (Advanced) Distributed Systems
Lecture 20

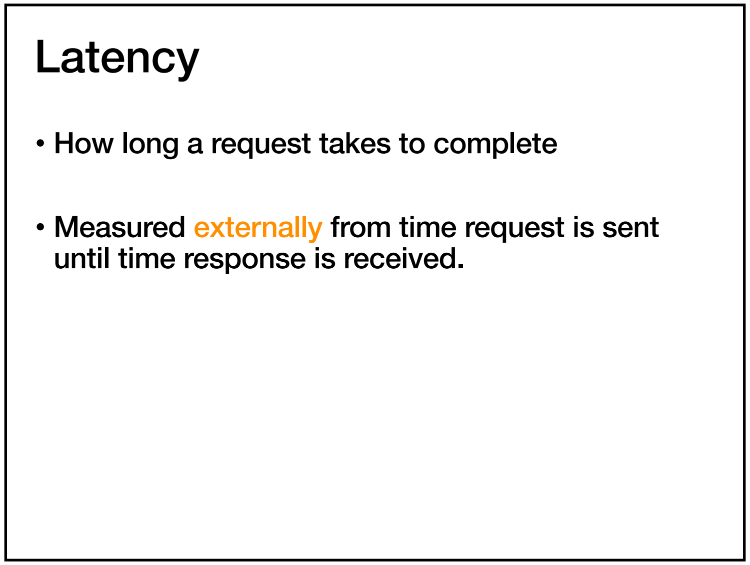
Mike Freedman & Wyatt Lloyd



Measuring Distributed Systems



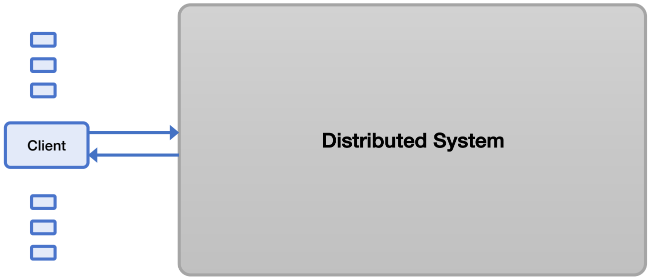
Measuring Distributed Systems



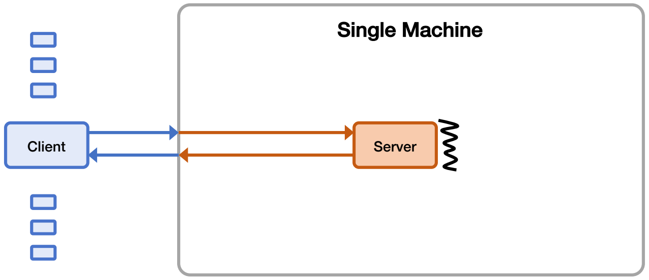
Latency

- How long a request takes to complete
- Measured externally from time request is sent until time response is received.

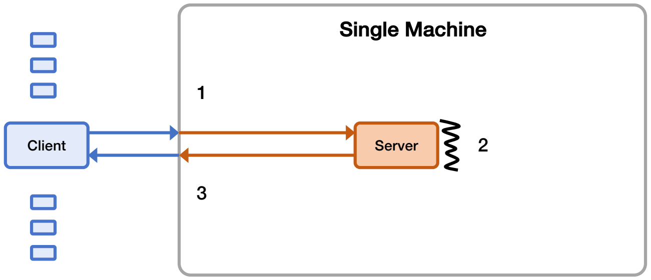
Latency, Measure Externally



Latency, Reason Internally



Latency, Reason Internally

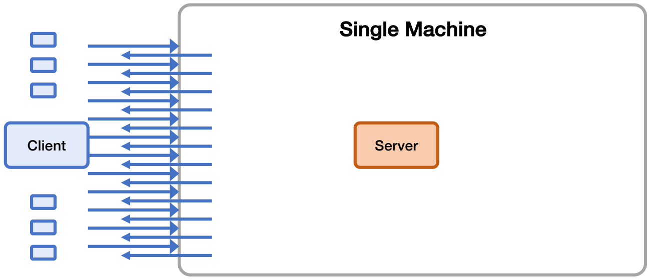


Latency = 1 + 2 + 3

Throughput

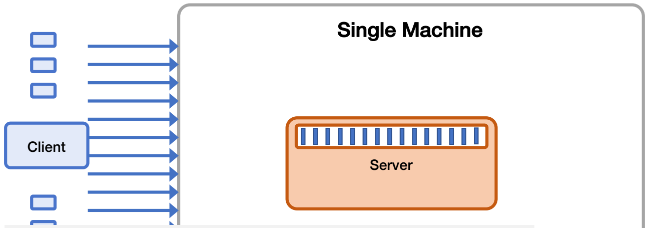
- How many operations per unit time a system can handle
  - Typically operations/second
- Measured externally as the rate that responses come out of the system

Max Throughput Example (Not Ideal)



$$\textbf{Throughput} = \frac{\text{Number of (valid) responses received by all clients}}{\text{End time – start time}}$$

Queuing Delay & Overload



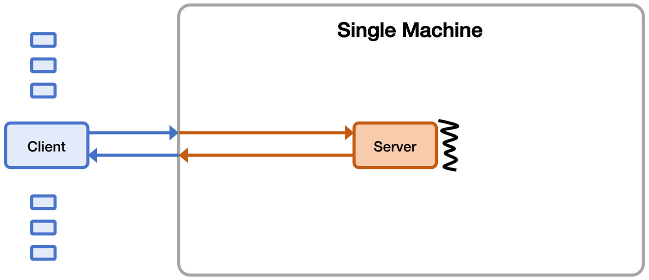
- **Queuing delay:** extra latency spent in queue(s)
- Higher load → increase in latency

- **Overload:** offered load > max system throughput
  - Queues get really long
  - Other weird/bad things happen
  - →Observed throughput < max system throughput

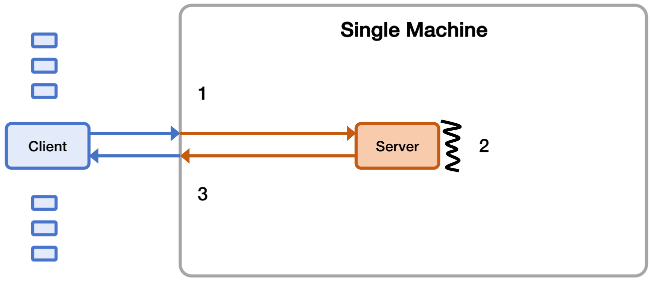
Measuring Throughput Method

1. Starting with low load
2. Increase load
3. Repeat until measured throughput stops increasing

Throughput, Reason Internally

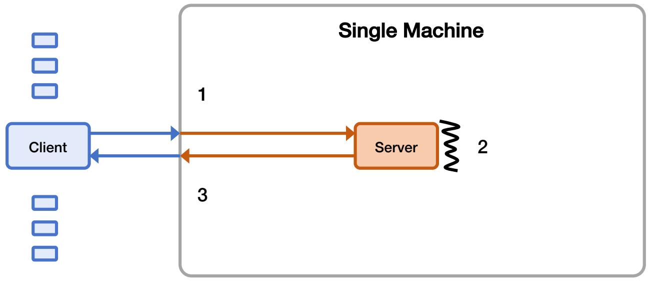


Throughput, Reason Internally



Throughput = min(1, 2, 3)

Throughput Bottlenecks (simplified)



Max throughput limited by some bottleneck resource:

1) Incoming bandwidth
2) Server CPU
3) Outgoing bandwidth

Load Generation

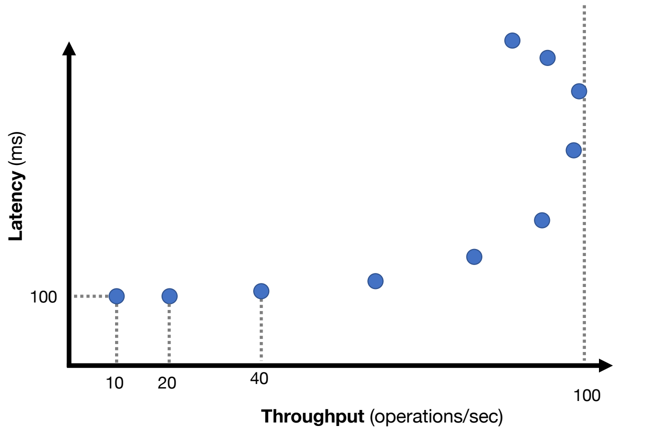
- Closed-loop
  - Each “client” sends one request, waits for the response to come back, and then sends another request
  - More “clients” => more load
- Open-loop
  - Load is generated independently of the response rate of the system, typically from a probability distribution
  - More directly control the load on the system
- Which one is more realistic?
- We’ll reason using closed-loop clients

Mental Experimental Setup

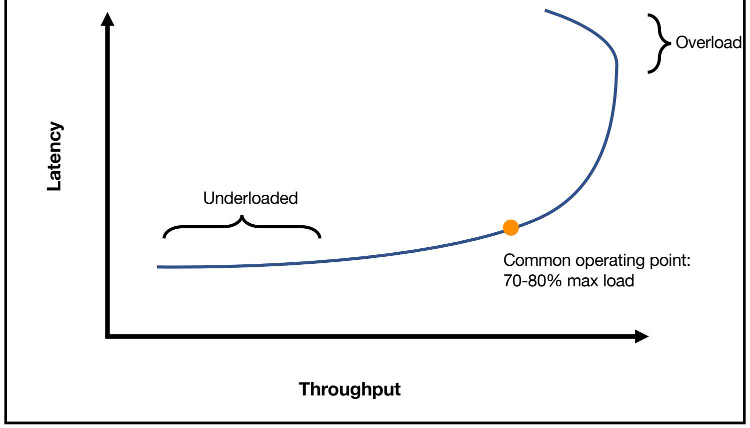
- Start with 1 closed-loop client
  - Expected latency?
  - Expected throughput?
- Double number of closed-loop clients
  - Expected increase in latency?
  - Expected increase in throughput?
- Repeat

Throughput-Latency Graph

Simple Setting: Single Server; Client-Server RTT 90ms;
Server Processing latency 10ms; Single-Threaded Server (100 ops/s)



Throughput-Latency Graph

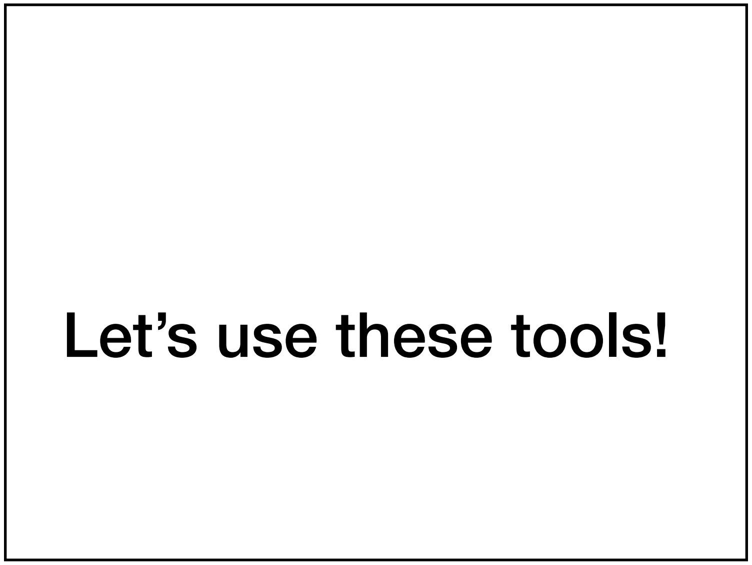


Throughput / Latency Relationship

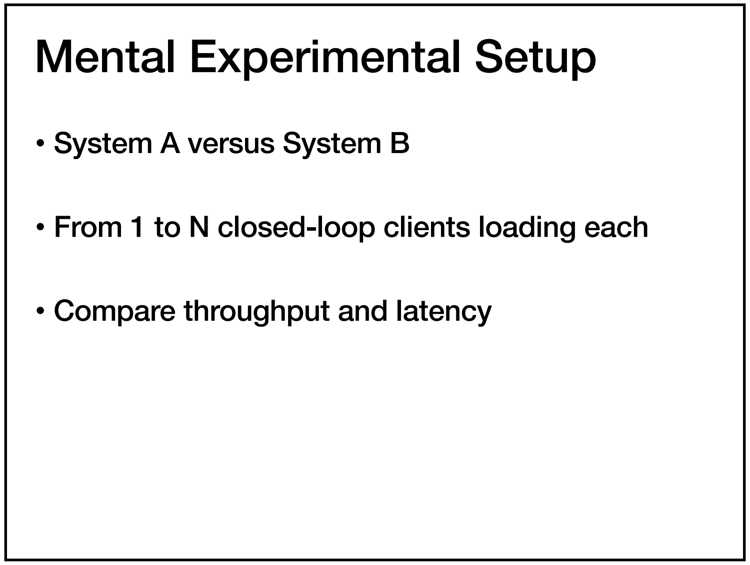
- Proportional at low load … but not high load
- Because measured throughput is a function of latency
  - i.e., throughput bottleneck is offered load
- Related, but you should reason about both
- For system A vs system B, all are possible:
  - A has lower latency and higher throughput than B
  - A has lower latency and lower throughput than B
  - A has higher latency and lower throughput than B
  - A has higher latency and higher throughput than B

Evaluation in Minutes not Months

- Reasoning using your mental model is much much faster than really doing it
- What would happen if?
  - I moved my servers from the San Jose datacenter to Oregon?
  - I switch from c5.xlarges to c5.24xlarges for my servers?
  - I doubled the number of servers?
  - I switch from system design X to system design Y?
    - replace single server with Paxos-replicated system?
    - replace Paxos with eventually consistent design?
    - add batching?
    - replace Paxos with new variant?

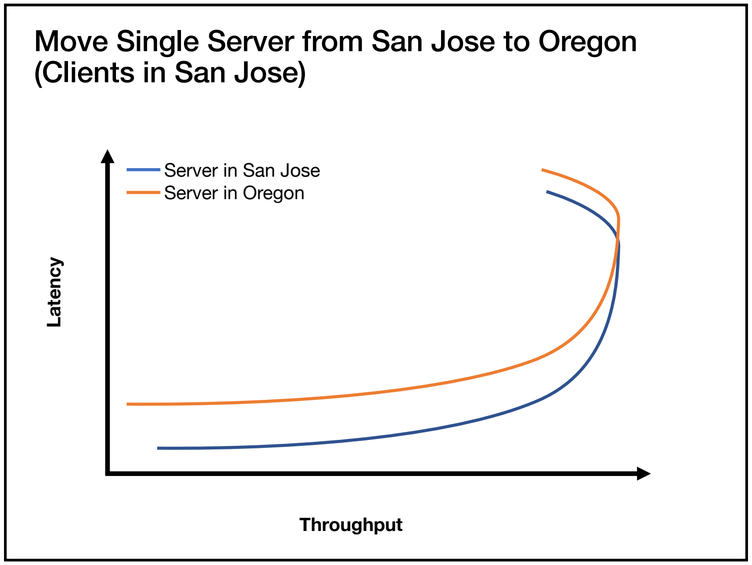


Let’s use these tools!

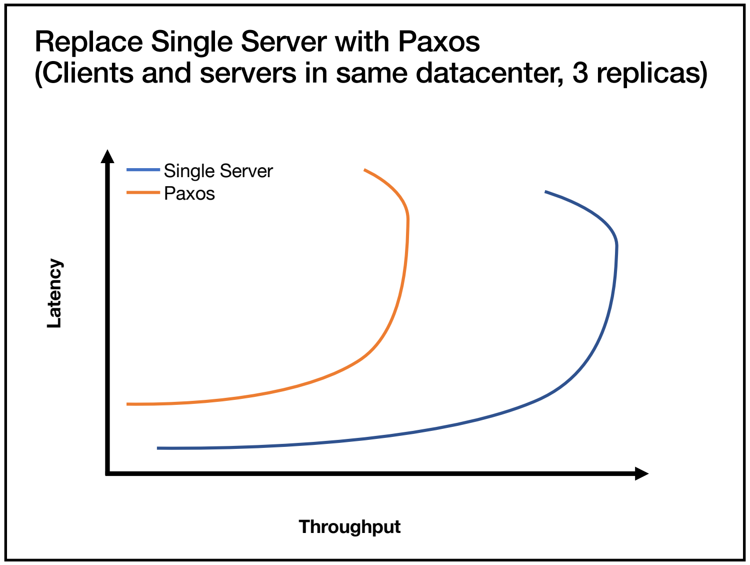


Mental Experimental Setup

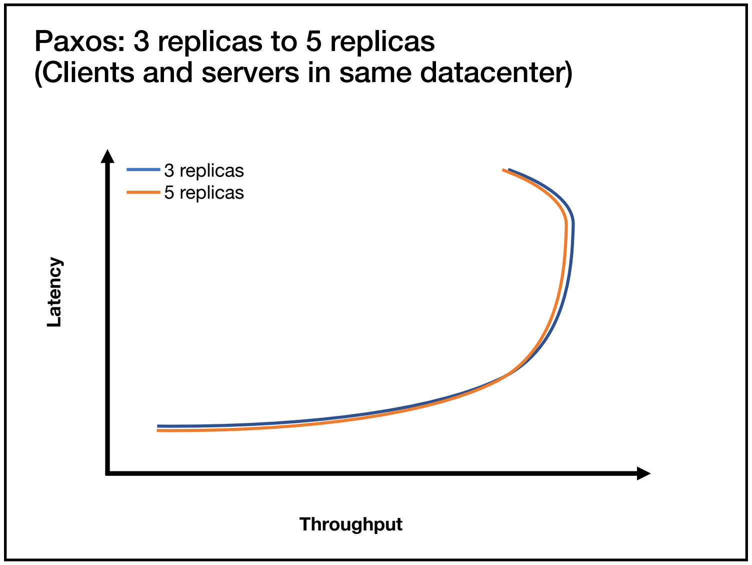
- System A versus System B
- From 1 to N closed-loop clients loading each
- Compare throughput and latency



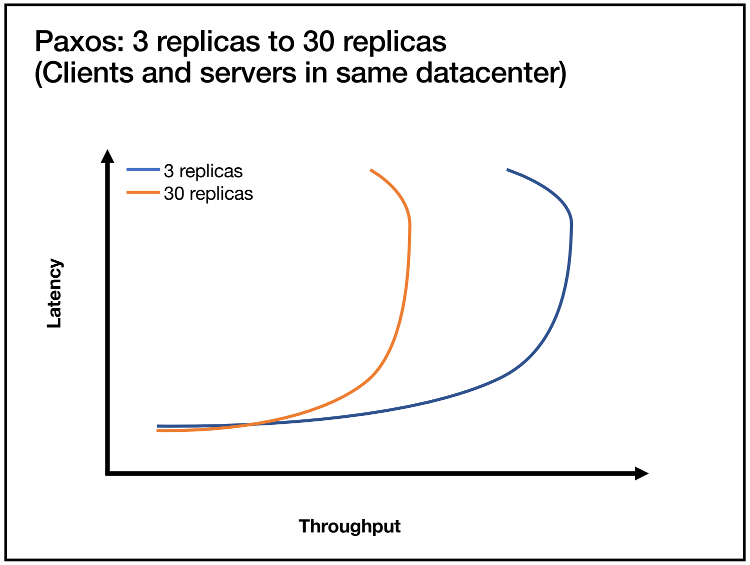
Move Single Server from San Jose to Oregon (Clients in San Jose)



Replace Single Server with Paxos (Clients and servers in same datacenter, 3 replicas)



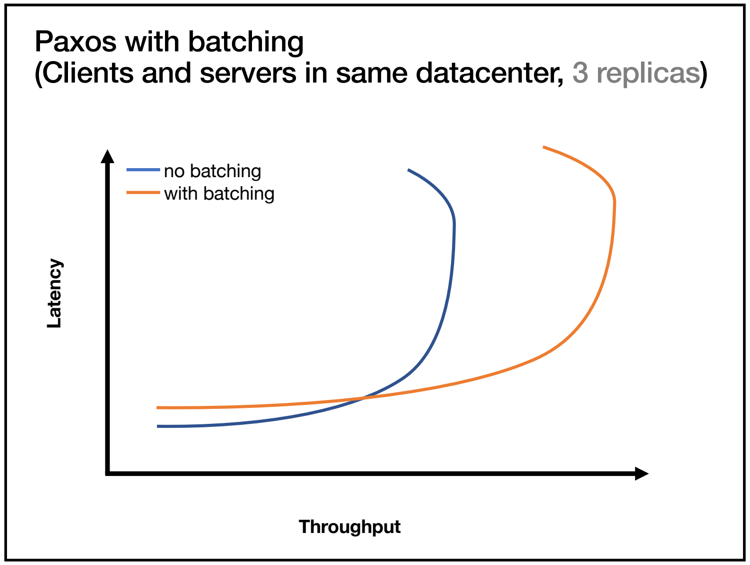
Paxos: 3 replicas to 5 replicas
(Clients and servers in same datacenter)



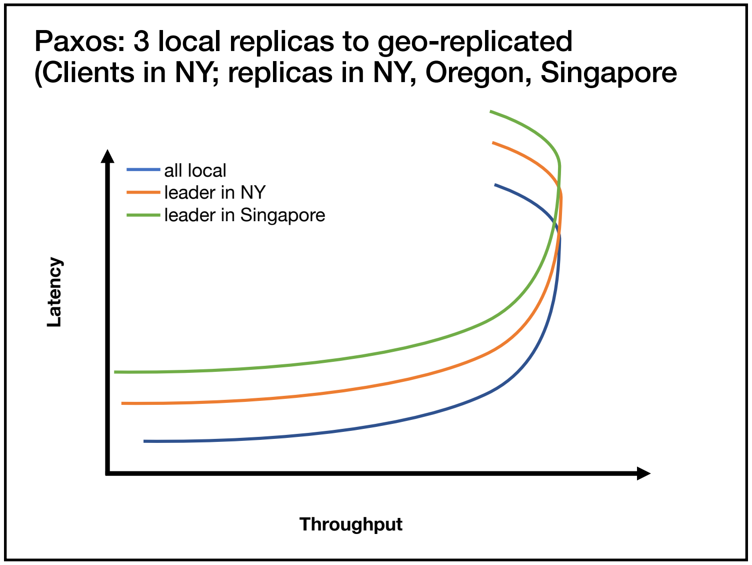
Paxos: 3 replicas to 30 replicas
(Clients and servers in same datacenter)

Batching

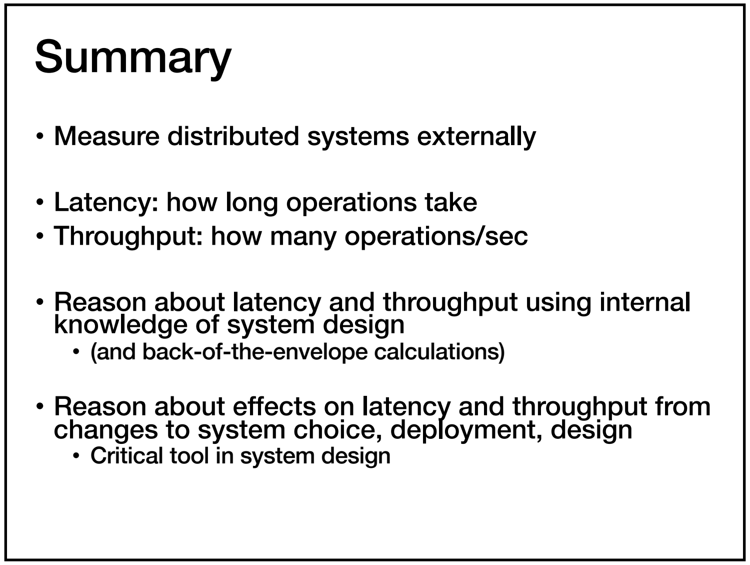
- Group together multiple operations
- Improves throughput, e.g.,
  - Marshall data together
  - Send to network layer together
  - Unmarshall data together
  - Handle group of operations together
- Delay processing/sending operations to increase batch size
  - Common way to trade an increase in latency for increase in throughput



Paxos with batching
(Clients and servers in same datacenter, 3 replicas)



Paxos: 3 local replicas to geo-replicated (Clients in NY; replicas in NY, Oregon, Singapore



Summary

- Measure distributed systems externally
- Latency: how long operations take
- Throughput: how many operations/sec
- Reason about latency and throughput using internal knowledge of system design
  - (and back-of-the-envelope calculations)
- Reason about effects on latency and throughput from changes to system choice, deployment, design
  - Critical tool in system design