Scalability

Oxford University
Software Engineering Programme
April 2021

scalability

/ˌskeɪləˈbɪlɪtɪ/

noun

1. the ability of something, esp a computer system, to adapt to increased demands

Collins English Dictionary - Complete & Unabridged 2012 Digital Edition

Speedup

- The **speedup** is defined as the performance of new / performance of old
  - e.g. move from 1 -> 2 servers
  - New system is 1.8 x faster than the old
    - In terms of transactions/sec (throughput)
  - Speedup = 1.8

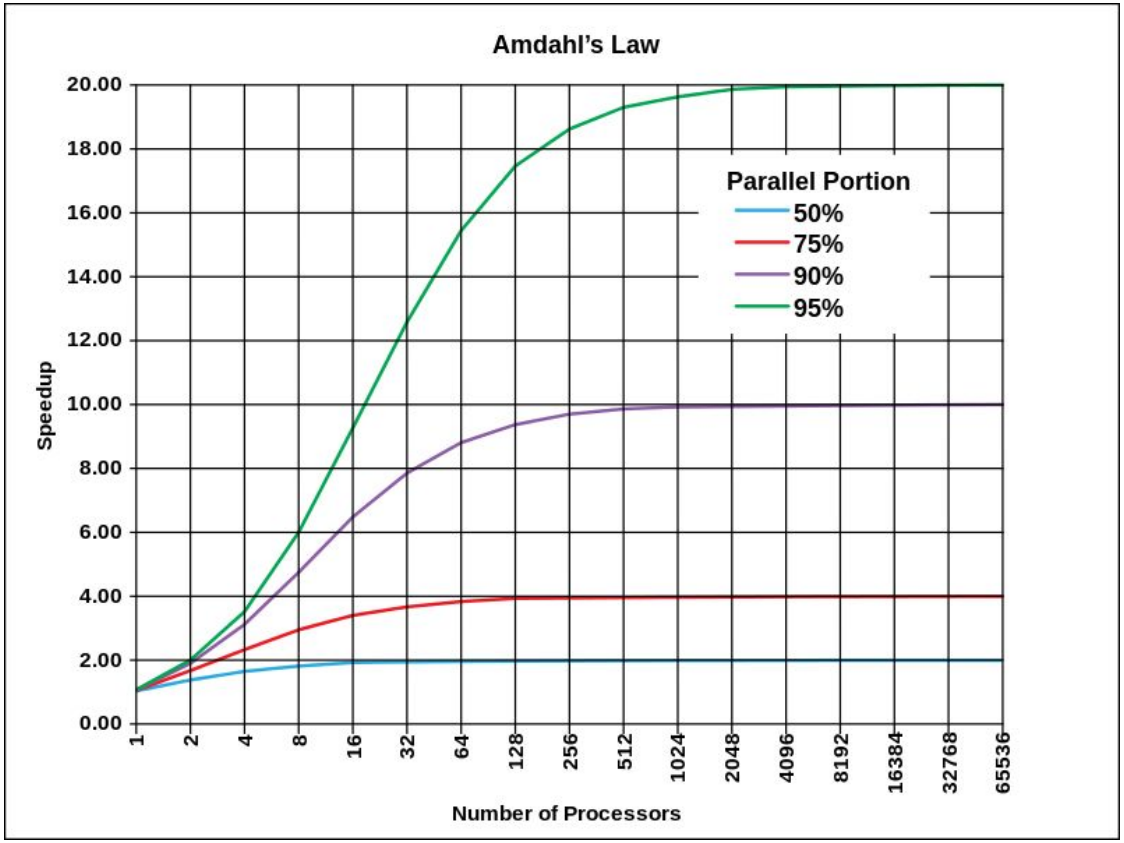
What inhibits speedup?

- In general you can split work into
  - Parallelizable and
  - Serial parts
- The serial parts stop you from scaling

Amdahl's Law

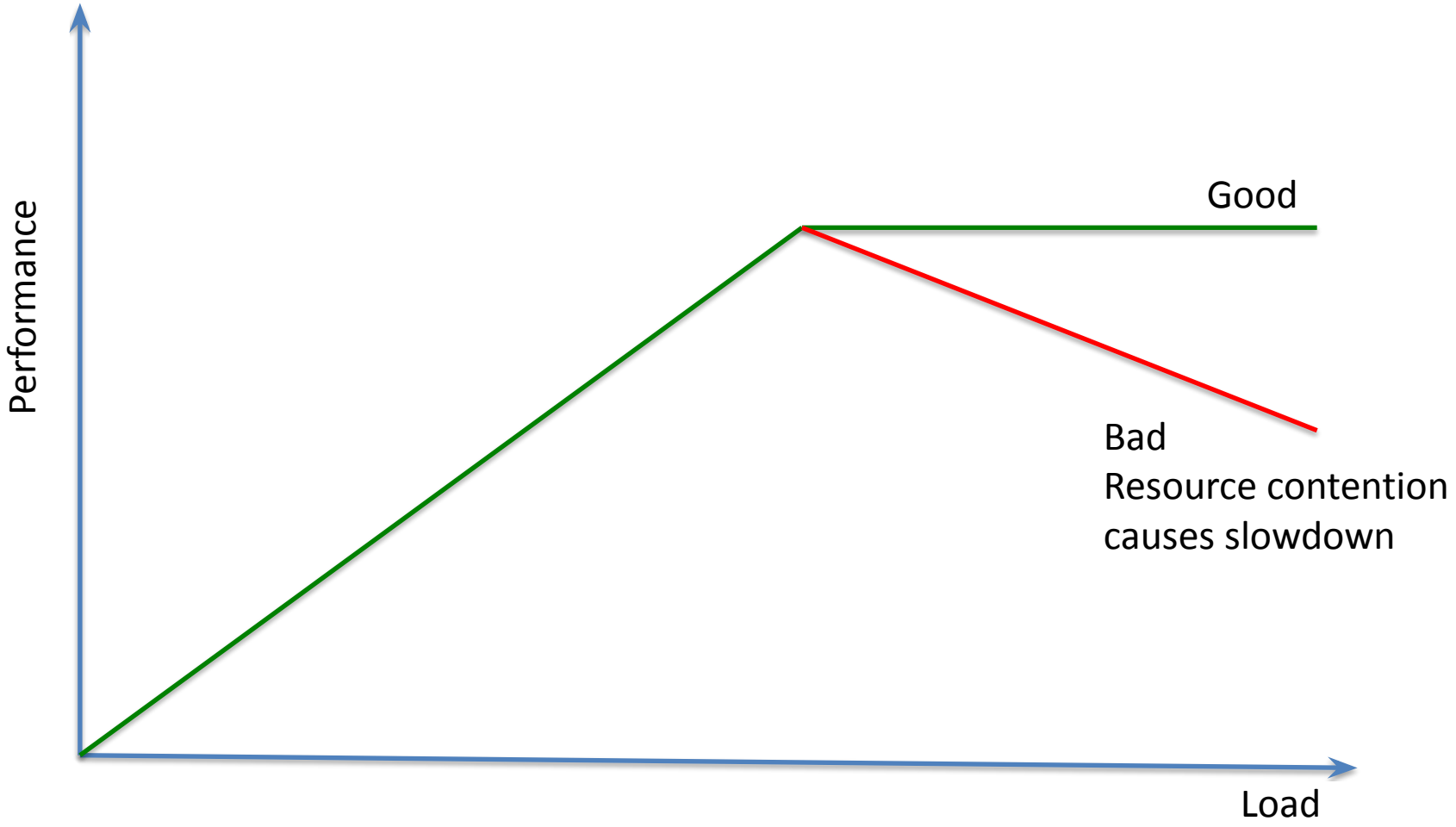
Theoretical speedup given a fixed data size

The speedup of a program using multiple processors in parallel computing is limited by the time needed for the serial fraction of the program, given a fixed size of data



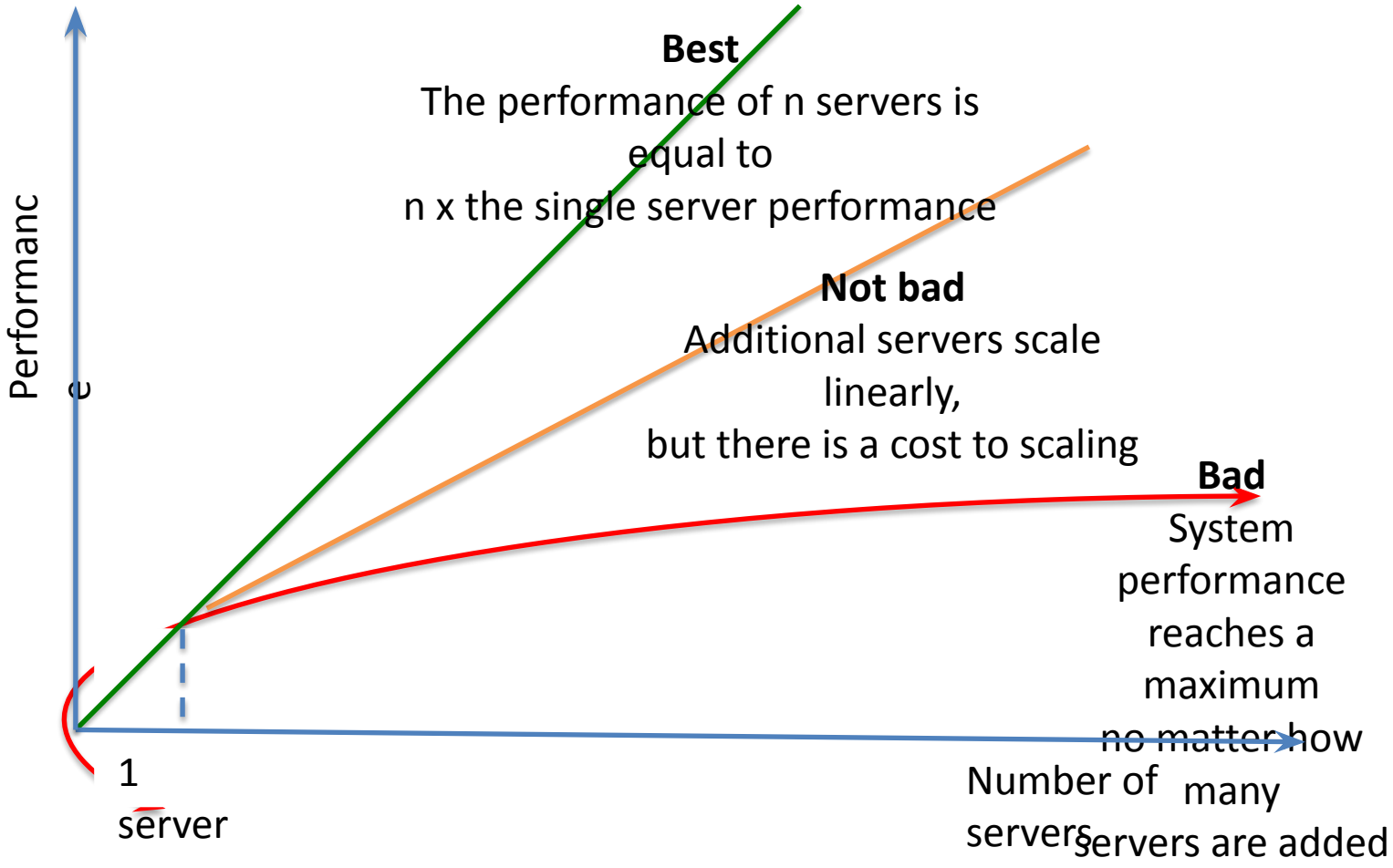
Performance

Single system under increasing load



Performance

Scaling servers when fully loaded



Karp-Flatt Metric

e is the Karp-Flatt Metric

ψ is the speedup

p is the number of processors

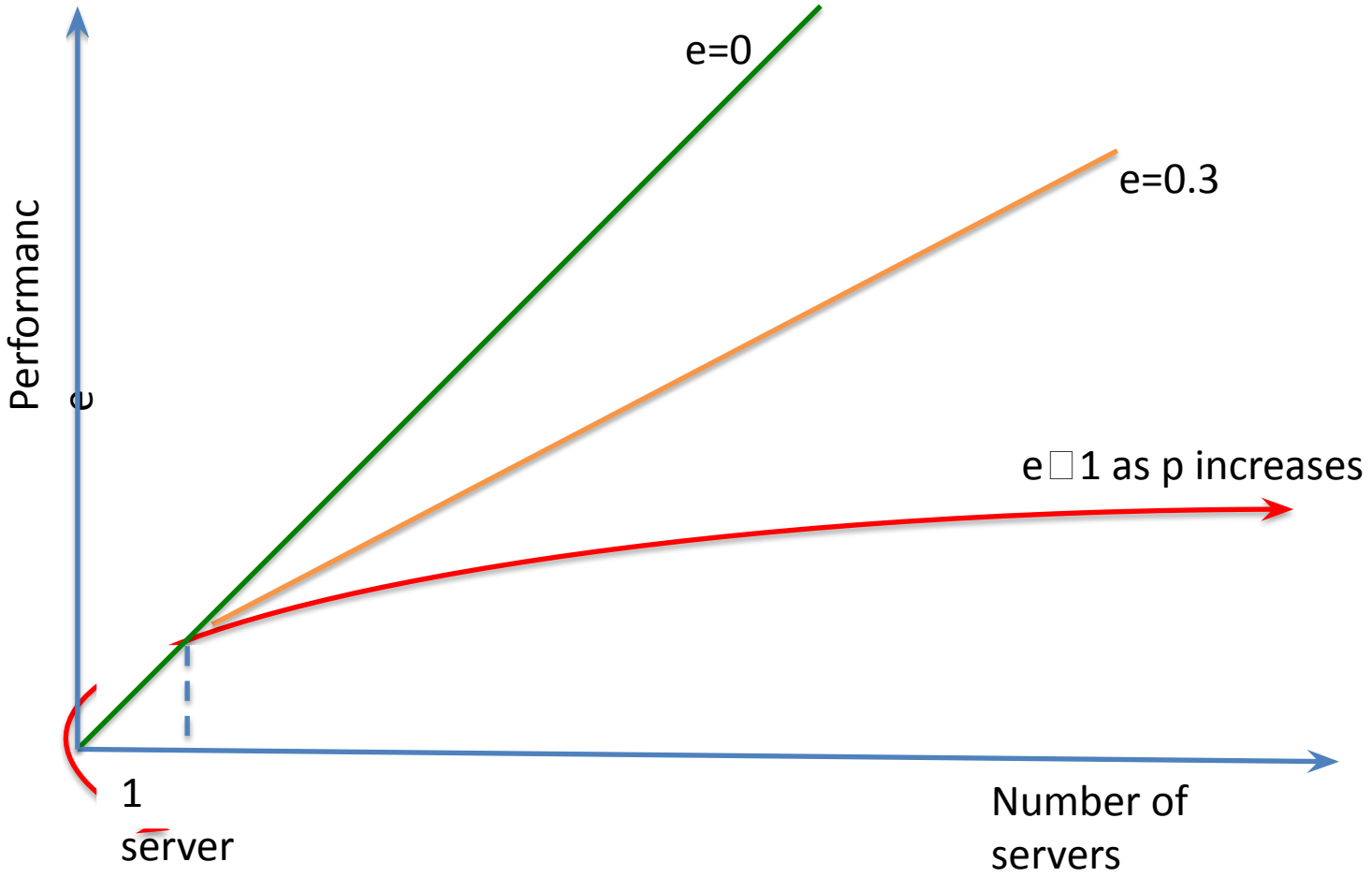
$$e = \frac{\frac{1}{\psi} - \frac{1}{p}}{1 - \frac{1}{p}}$$

e = 0 is the best

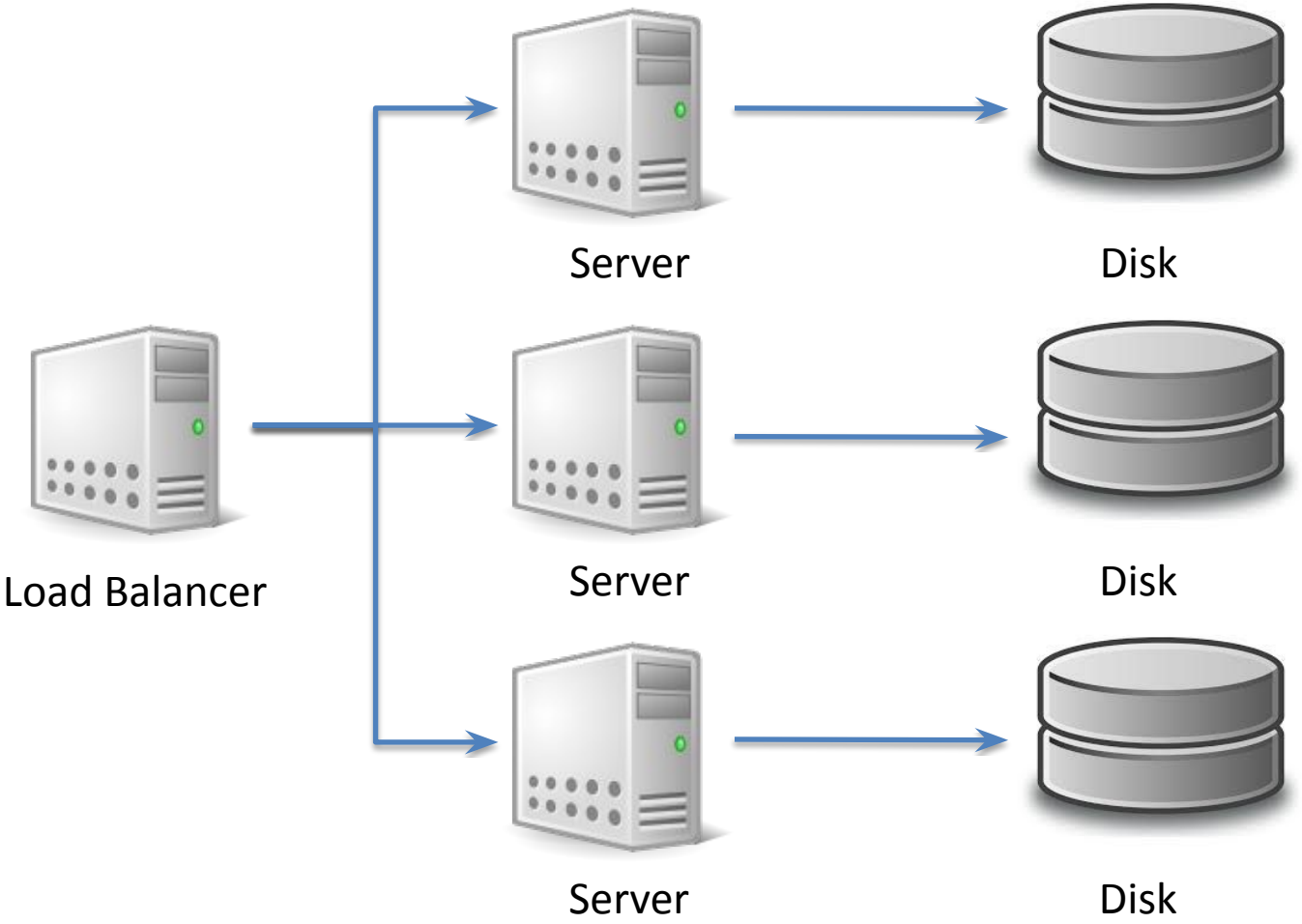
e = 1 indicates no speedup

e > 1 indicates adding processors slows down the system!!!

Karp-Flatt metric



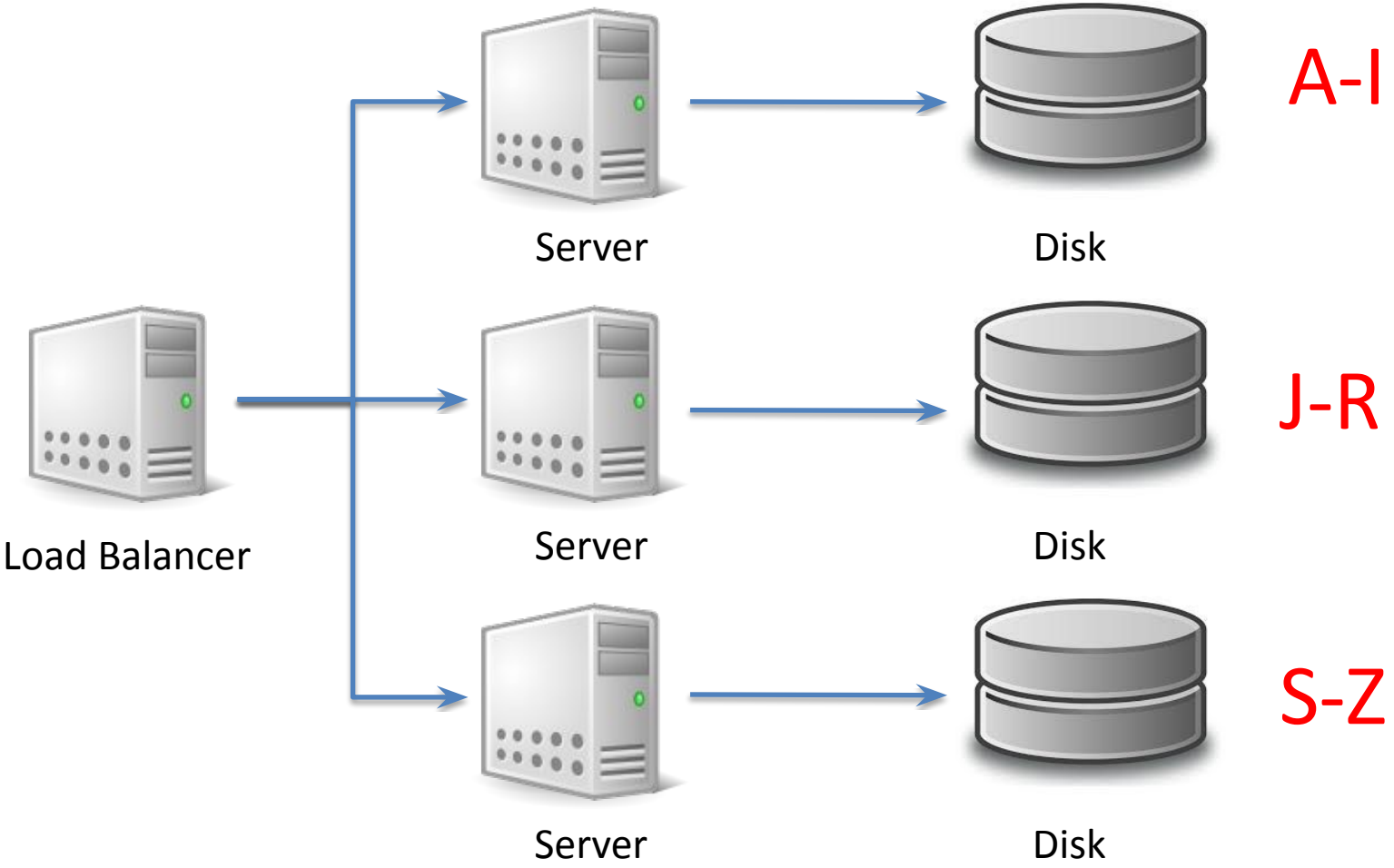
Shared Nothing Architecture



Shared Nothing Architecture

- Implies there is no serial part to the computation
- Karp-Flatt Metric of 0
  - Assuming 100% efficient load balancing
- In practice, this is difficult!

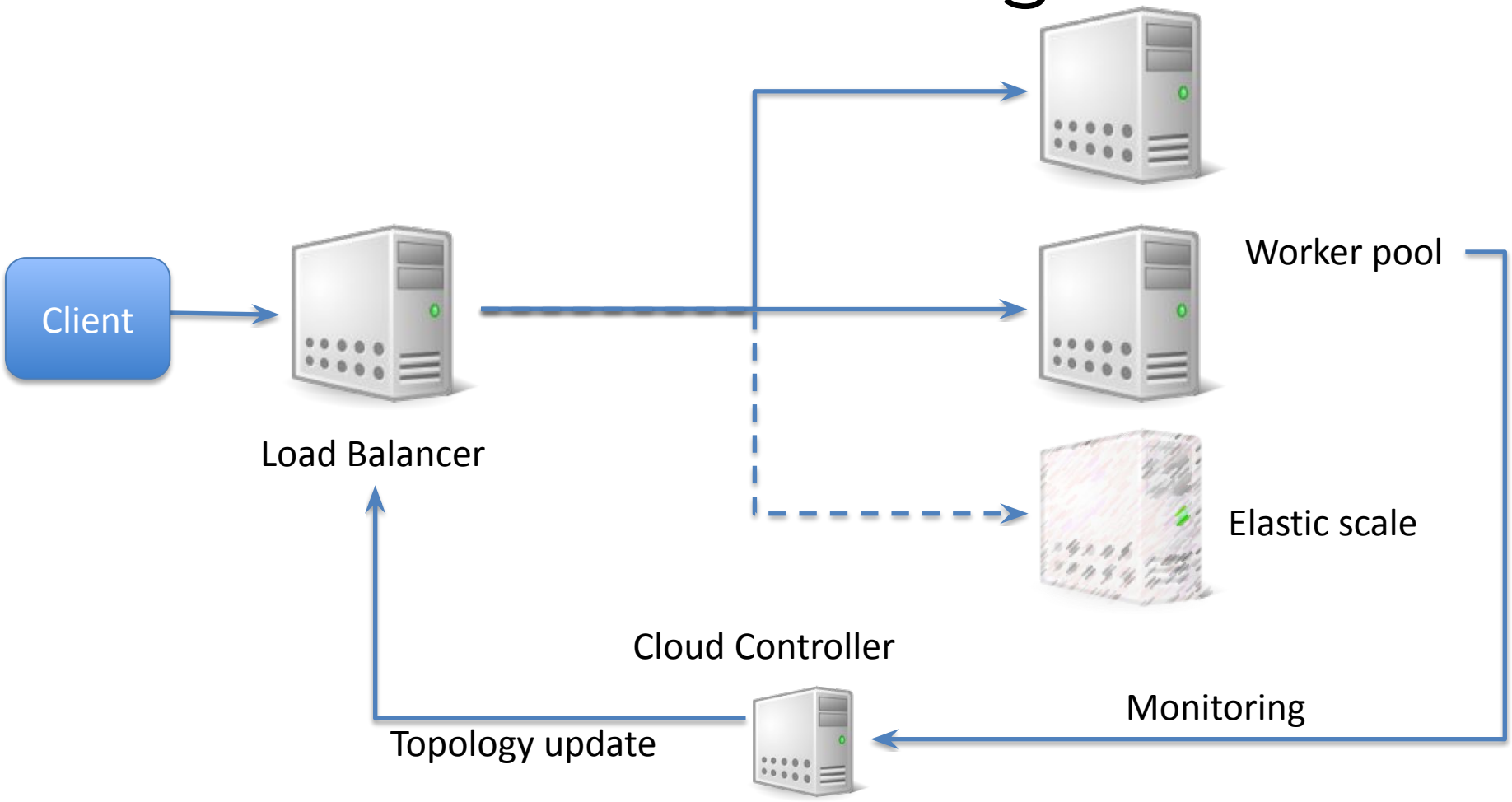
Partitioning / Sharding



Problems with Sharding

- Imbalance
  - Fewer S-Z's than A-I's
- Failover
- Adding new servers requires a re-balance
  - Is this automatic or manual?!

Load Balancer-based elastic scaling



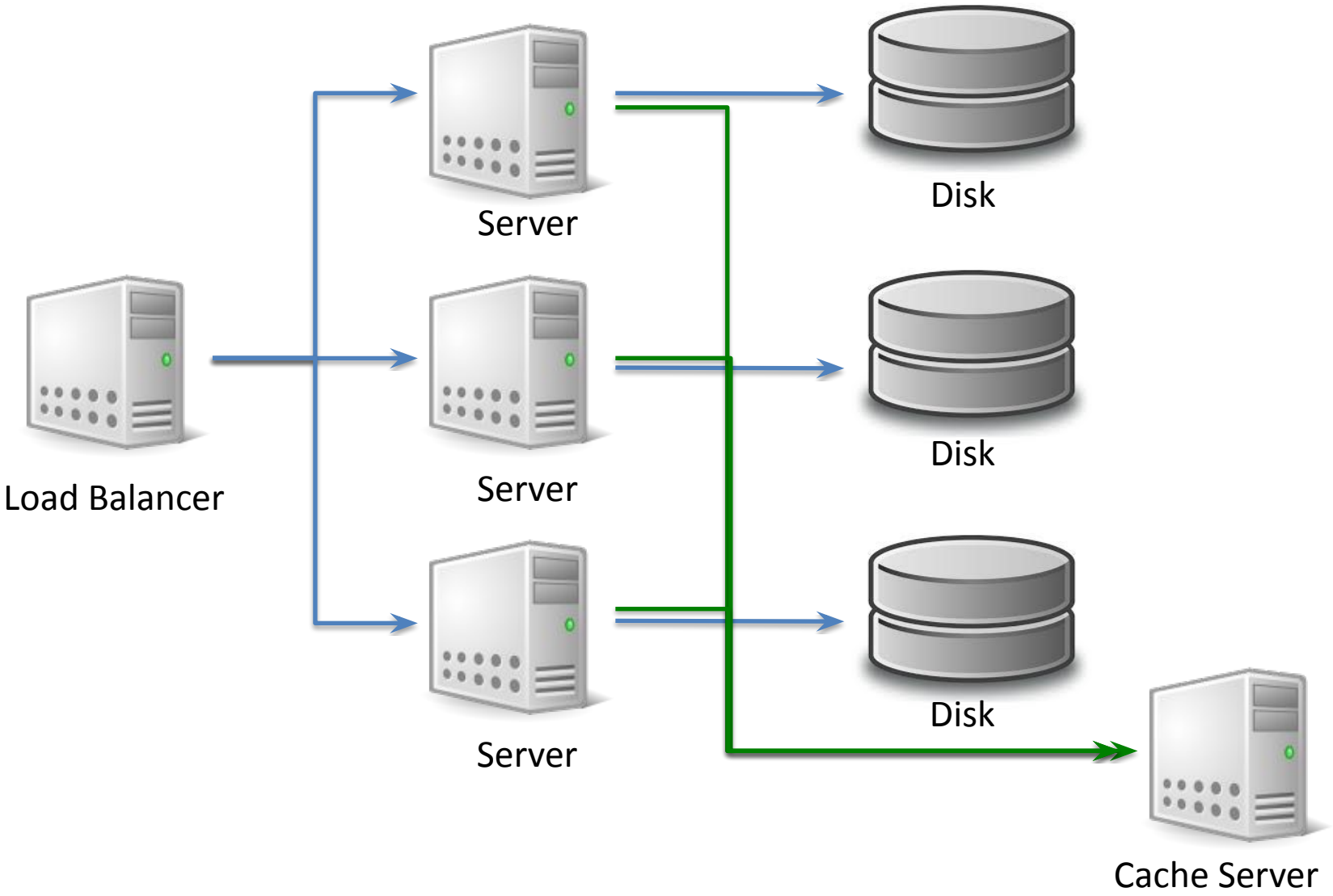
Statelessness is hard

- There is a lot of intermediate calculation in most web systems that needs to be stored between transactions.
- The exemplar is the shopping cart

Intermediate state storage

- In-memory
  - Breaks statelessness, limits scalability
- In the client
  - OK for some APIs but can be slow and hard to program
- In the database
  - Slow and expensive
- Cache servers:
  - The client keeps a cookie, which is the key to the datastore
  - The usual practise
  - Redis, memcached, Hazelcast, Infinispan

Cache


Server
Disk
Load Balancer
Server
Disk
Server
Disk
Cache Server

Questions?