Scalability

Principles of Functional Programming

Roland Kuhn

Scalability

Low performance means the system is slow for a single client.

Low scalability means the system is fast when used by a single client but slow when used by many clients.

Replication of Actors

One actor can process one message at a time.

Stateless replicas can run concurrently.

Replication of Actors

One actor can process one message at a time.

Stateless replicas can run concurrently.

Routing messages to worker pools:

- stateful (round robin, smallest queue, adaptive, ...)
- stateless (random, consistent hashing, ...)

Round-Robin Routing

- equal distribution of messages to routees
- hiccups or unequal message processing times introduce imbalance
- imbalances lead to larger spread in latency spectrum

Smallest Mailbox Routing

- requires routees to be local to inspect message queue
- evens out imbalances, less persistent latency spread
- high routing cost, only worth it for high processing cost

Shared Work Queue

- requires routees to be local
- most homogenous latency
- effectively a pull model

Adaptive Routing

- requires feedback about processing times, latencies, queue sizes
- feedback can be sampled coarsely
- steering the routing weights subject to feedback control theory
  - oscillations
  - over-dampening

Random Routing

- asymptotically equal distribution of messages to routees
- no shared state necessary: low routing overhead
- works with several distributed routers to the same routees
- can stochastically lead to imbalances

Consistent Hashing

- splitting incoming message stream according to some criterion
- bundle substreams and send them to the same routees consistently
- can exhibit systematic imbalances based on hashing function
- different latencies for parts of the input spectrum
- no shared state necessary between different routers to the same targets

Replication of Stateful Actors

Consistent Hashing can be used if substreams correspond to independent parts of the state.

Replication of Stateful Actors

Consistent Hashing can be used if substreams correspond to independent parts of the state.

Multiple writers to the same state require appropriate data structures and are eventually consistent.

Replication of Stateful Actors

- based on persisted state
- only one instance active at all times
- consistent routing to the active instance
- possibly buffering messages during recovery
- migration means recovery at a different location

Summary

Asynchronous message passing enables vertical scalability.

Location transparency enables horizontal scalability.