Building Scalable Stateful Services
Stateless Services
Stateless Services
Stateless Services
Stateless Services
Stateless Services
Stateless Services

---

Service

Service

Service

---

Database

---

Service

Service

Service
Stateless Services

Service

Service

Service

Database
Stateless Services

Service
Service
Service
Stateless Services
Stateless Services
Stateless Services

Diagram showing multiple phones connected to services and a database.
Data Shipping Paradigm
Data Shipping Paradigm
Data Shipping Paradigm
Data Shipping Paradigm

[Diagram showing a phone connected to services and then to a database]
Data Shipping Paradigm

[Diagram showing a mobile device connected to services, with data flowing to and from a database.]
Data Shipping Paradigm
Data Shipping Paradigm
Data Shipping Paradigm
Data Shipping Paradigm
Data Shipping Paradigm
Data Shipping Paradigm
Data Shipping Paradigm
Data Shipping Paradigm

[Diagram showing a mobile device connected to services with a database]

- Service
- Service
- Service
Data Shipping Paradigm
Data Shipping Paradigm
Data Shipping Paradigm
Data Shipping Paradigm

Front End

Service

Service

Service

Service

Service

Service
Data Shipping Paradigm
Data Shipping Paradigm
Data Shipping Paradigm

Front End

Service

Service

Service

Service

Service

Service

Service

Service
Overview

- Benefits
- Building
- Real World
- Caution
Data Locality
For Low Latency & Data Intensive Services
Function Shipping Paradigm
Function Shipping Paradigm
Function Shipping Paradigm
Function Shipping Paradigm
Function Shipping Paradigm
Function Shipping Paradigm
Function Shipping Paradigm
Function Shipping Paradigm
Function Shipping Paradigm
Function Shipping Paradigm
Function Shipping Paradigm
Data Shipping Paradigm
Data Shipping Paradigm
Data Shipping Paradigm
Just Put A Cache On It?
Concurrent Issues

Greater Operational Burden
Sticky Connections & Consistency

Additional Available Consistency Models
Sticky Connections
Sticky Connections
Consistency Models

- Linearizable
  - Sequential
    - Causal
      - Pipelined Random Access Memory
        - Read Your Write
      - Write From Read
      - Monotonic Read
      - Monotonic Write

- CP Consistency
- AP Consistency
Consistency Models

- Linearizable
  - Sequential
    - Causal
      - Pipelined Random Access Memory
        - Read Your Write
      - Monotonic Read
      - Monotonic Write
      - Write From Read

Consistency Models:
- CP Consistency
- AP Consistency
- AP Consistency w/ Sticky Connections
“Whether or not read-your-write, session and monotonic consistency can be achieved depends in general on the "stickiness" of clients to the server that executes the distributed protocol for them... Using sessions, which are sticky, makes this explicit and provides an exposure level that clients can reason about.”

- Werner Vogel 2007
Building Sticky Connections

For Low Latency & Data Intensive Services
Building Sticky Connections
Building Sticky Connections
Persistent Connections

Problems
- Load Balancing Problems
- No Stickiness Once Connection Breaks
Persistent Connections

Problems
Load Balancing Problems
No Stickiness Once Connection Breaks
Routing Logic

Problems to Solve

- Cluster Membership
- Work Distribution
Routing Logic

Problems to Solve

- Cluster Membership
- Work Distribution
Routing Logic

Problems to Solve

- Cluster Membership
- Work Distribution
Cluster Membership

Static  Dynamic
Static Cluster Membership

- Machine Outage means Partial Service Outage
- Downtime to Add Capacity
Static Cluster Membership

- Machine Outage means Partial Service Outage
- Downtime to Add Capacity
Static Cluster Membership

- Machine Outage means Partial Service Outage
- Downtime to Add Capacity
Dynamic Cluster Membership

- Fault Tolerance
Dynamic Cluster Membership

- Fault Tolerance
Dynamic Cluster Membership

Fault Tolerance
Dynamic Cluster Membership

Fault Tolerance
Dynamic Cluster Membership

○ Fault Tolerance ○
Dynamic Cluster Membership

Gossip Protocols vs Consensus Systems

Availability vs Consistency
Consensus Systems for Consistency

Zookeeper | Paxos | ETCD | Raft
Gossip Protocols for Availability

- Assumes Non-Reliable Networks
- Information Dissemination
- Pairwise Communication
Gossip Protocols for Availability

- Assumes Non-Reliable Networks
- Information Dissemination
- Pairwise Communication
Gossip Protocols for Availability

- Assumes Non-Reliable Networks
- Information Dissemination
- Pairwise Communication
Gossip Protocols for Availability

- Assumes Non-Reliable Networks
- Information Dissemination
- Pairwise Communication
Gossip Protocols for Availability

- Assumes Non-Reliable Networks
- Information Dissemination
- Pairwise Communication
Gossip Protocols for Availability

- Assumes Non-Reliable Networks
- Information Dissemination
- Pairwise Communication
Gossip Protocols for Availability

- Assumes Non-Reliable Networks
- Information Dissemination
- Pairwise Communication
Gossip Protocols for Availability

- Assumes Non-Reliable Networks
- Information Dissemination
- Pairwise Communication
Gossip Protocols for Availability

- Assumes Non-Reliable Networks
- Information Dissemination
- Pairwise Communication
Work Distribution

Random Placement  Consistent Hashing  Distributed Hash Tables
Random Placement

Write Anywhere

Read from Everywhere
Random Placement

Write Anywhere

Read from Everywhere
Random Placement

Write Anywhere

Read from Everywhere
Random Placement

Write Anywhere

Read from Everywhere
Random Placement

Write Anywhere

Read from Everywhere
Random Placement

Write Anywhere
Read from Everywhere
Consistent Hashing & Random Trees: Distributed caching protocols for relieving hot spots on the World Wide Web
Consistent Hashing & Random Trees: Distributed caching protocols for relieving hot spots on the World Wide Web
Consistent Hashing & Random Trees: Distributed caching protocols for relieving hot spots on the World Wide Web
Distributed Hash Table

Non-Deterministic Placement

Node A
Node B
Node C
Distributed Hash Table

Non-Deterministic Placement

Node A
Node B
Node C
Distributed Hash Table

Non-Deterministic Placement
Distributed Hash Table

Non-Deterministic Placement
Distributed Hash Table

Non-Deterministic Placement

Node A
Node B
Node C
Scuba is a fast, scalable, distributed, in-memory database built at Facebook. It is the workhorse behind code regression analysis & bug report, revenue, and performance debugging.

Fan-out request to all machines in the cluster

Compose Results

Return Results and Completeness
Scuba is a fast, scalable, distributed, in-memory database built at Facebook. It is the workhorse behind code regression analysis & bug report, revenue, and performance debugging.

Fan-out request to all machines in the cluster

Compose Results

Return Results and Completeness
**Scuba** is a fast, scalable, distributed, in-memory database built at Facebook. It is the workhorse behind code regression analysis & bug report, revenue, and performance debugging.

Fan-out request to all machines in the cluster

Compose Results

Return Results and Completeness
Uber Ringpop is an open-source Node.js library that brings application-layer sharding to many of their dispatching platform services.

Swim Gossip Protocol + Consistent Hashing
Uber Ringpop is an open-source Node.js library that brings application-layer sharding to many of their dispatching platform services.

Swim Gossip Protocol + Consistent Hashing
Uber Ringpop is an open-source Node.js library that brings application-layer sharding to many of their dispatching platform services.

Swim Gossip Protocol + Consistent Hashing
Orleans is a runtime and Programming model for building distributed systems based on the Actor Model from the eXtreme Computing Group at MSR.

Gossip Protocol
+ Consistent Hashing
+ Distributed Hash Table
Orleans is a runtime and Programming model for building distributed systems based on the Actor Model from the eXtreme Computing Group at MSR.
Orleans is a runtime and Programming model for building distributed systems based on the Actor Model from the eXtreme Computing Group at MSR.

Gossip Protocol + Consistent Hashing + Distributed Hash Table
Orleans Distributed Hash Table
Orleans Distributed Hash Table

Orleans Cluster

[Diagram showing a mobile device connected to an Orleans Cluster with distributed hash table entries]
Orleans Distributed Hash Table
Orleans Distributed Hash Table
Orleans Distributed Hash Table
Orleans Distributed Hash Table

Orleans Cluster

[Diagram of Orleans Cluster with a smartphone pointing to a specific server]
Orleans Distributed Hash Table
Orleans Distributed Hash Table
Caution
Stateful Services Ahead
Unbounded Data Structures

Implicit Assumptions are the Killer of Distributed Systems
Memory Management
Get Ready to Make Friends with the Garbage Collector Profiler
Reloading State

- First Connection
- Recovering From Crashes
- Deploying New Code
Fast Restarts at Facebook

“Our Key Observation is that we can decouple the memory lifetime from the process lifetime. When we shutdown a server for a planned upgrade.”
Make Assumptions Explicit
for Reliable Distributed Systems
Conclusion

Data Locality & Available Consistency

Cluster Membership & Work Distribution

Successful Stateful Real World Systems

Caution: Some New Challenges
Questions

https://github.com/CaitieM20/ScalingStatefulServices

@Caitie