

Heterogeneous backend architecture at Ninchat

Timo Savola Backend Lead, Ninchat

Reaktor Dev Day 2013-10-04



. IRC

- Connecting things
- . Need a solid platform



- Generic messaging
 - Text-based chat for humans
 - Arbitrary message types for robots
- Public and private channels
- One-on-one conversations



- Guest users
 - Temporary, anonymous chatters
- Permanent users
 - Verified user identity
 - Create channels etc.



Open API

- WebSocket or HTTP long-polling
- JSON headers
- Binary payloads
- https://ninchat.com/api
- Client libraries
 - https://github.com/ninchat/ninchat-python
 - https://github.com/ninchat/ninchat-java



- Official clients
 - https://ninchat.com
 - Android app (beta)
- Embedding for third-party sites
 - Customer service
 - Shoutbox



Flexibility

- Business needs
- . Low maintenance overhead
 - Automatic configuration
 - Automatic failover



- Long client connections
- Asynchronous event delivery
 - One-to-many
- . Data model
 - Object graph consistency
 - Immutable content









- Manage client connections
- Hide the complexity of HTTP longpolling
- High concurrency
- . Stateless



- . Implements all Ninchat features
- Request/broadcast programming model
- Database access
- Fast development pace
- . Stateless



- Live session registry
- Buffer events in memory
 - o To be consumed by clients...

....with bad connections...

- ...so keep them until acknowledged
- . Resembles Redis a bit
- Minimal responsibilities
- Stateful



Logic workers

Connection servers

Session nodes



Logic workers

Connection servers

Session nodes



Logic workers

Connection servers

Session nodes





- Goroutines for concurrency
 - Segmented stacks
- . Communicate, don't synchronize
- . Fresh take on interfaces
 - Implicit satisfaction
- . Good WebSocket support early on
- . 2009, Google
- http://golang.org

Python for workers

- Rapid application development
- Massive library support
- Flexible language comes with high execution overhead



- Predictable execution
 - Performance
 - Memory allocation
- Single-threaded implementation
 - Less bugs
 - Concurrent updates would just contend
- . Less agile language



Python Dynamic typing Go

Static type inferenceCompile time duck interfacing

C++

Static typing



Python Context manager protocol • Go **Deferred function calls** • C++• RAII







- Message queue library
- No central broker required
- High-level I/O patterns, including:
 - Multi-sender pipeline
 - Request/response multiplexing
- Asynchronous API
 - Simplifies application code
- http://zeromq.org



- In-house protocol
- One-way message passing
- Load-balancing
 - Untargeted
- Aggressive failover
- Inline health check
 - o Did the first database read work?
- https://github.com/ninchat/offhand



Clients

Connect



Send requests

Servers





Protocol Buffers

- Object serialization format
- . Generates code from a "schema"
- Same data object model in all
 - languages
- Extend with backward-compatibility
- . Google
- https://code.google.com/p/protobuf



- Protocol Buffers definitions
 - Compact
 - Convenient
- . ZeroMQ pipeline
 - No blocking



- Plug into DNS via dnsmasq
- ZeroMQ broadcasting
 - Edge-triggered state changes
- Amazon S3
 - Highly available configuration storage
 - Level-triggered state changes
- https://github.com/ninchat/nameq



- . AWS
- PostgreSQL
- Python 2.7
- Ubuntu LTS



timo@ninchat.com