



advanced k/v store ➡ data structure server

disk-backed, in-memory store

is Redis a NOSQL DB, a cache or a messaging server?

Redis as queue server

Lists

- lists of binary-safe strings
- push/pop on both ends: can be used as queues
- LPUSH + RPOP / RPUSH + LPOP
- BRPOP/ BLPOP blocks until items are available
- blocking queue means queue without polling

Redis as pubsub server

Publish / Subscribe

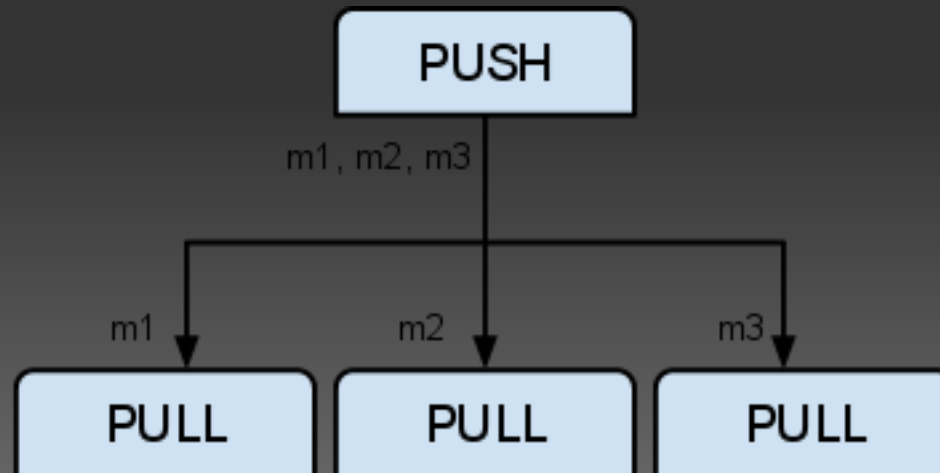
- does not involve key/value storage
 - realtime, channeled messaging
 - (UN)SUBSCRIBE to one or more channels
 - P(UN)SUBSCRIBE to channels matching pattern
 - PUBLISH message to channel
-
- buffering for slow connected clients (warning!)

ØMQ \zeromq\

- messaging client library
- support most messaging patterns
 - request/reply, pub/sub, push/pull, pair
- serverless, no intermediate broker
- simple API, similar to sockets
- blazing fast! orders of magnitude faster than most AMQP systems
- TCP, MULTICAST, IPC, INPROC transports
- originally the zero in ØMQ was meant as "zero broker" and "zero latency"

ØMQ PUSH/PULL

- pipelining: connect nodes in a fan-out / fan-in pattern
- load balancing/fair queuing
- parallel tasks distribution



ØMQ PUB/SUB

- connect set of publishers to set of subscribers
- loosely coupled
- *volatile* published stream
- publish on topic
- subscribe on topics / topic patterns

