ZooKeeper

System goals

- Writing a distributed system is difficult
- Coordinating a cluster can be frustrating and full of bugs
- ZooKeeper: “high performance coordination service”

System overview

- Written in Java
- Data type
  - Abstract nodes called *znode*
    - Znodes arranged in a tree
    - Znodes can store data and have children nodes
- Watches: each read can add a watch for a znode
- ACLs

Metrics

- Consistency
    - Sequential consistency: updates from a client will be applied in the order that they were sent
    - Atomicity: updates either fail or succeed, no partial updates

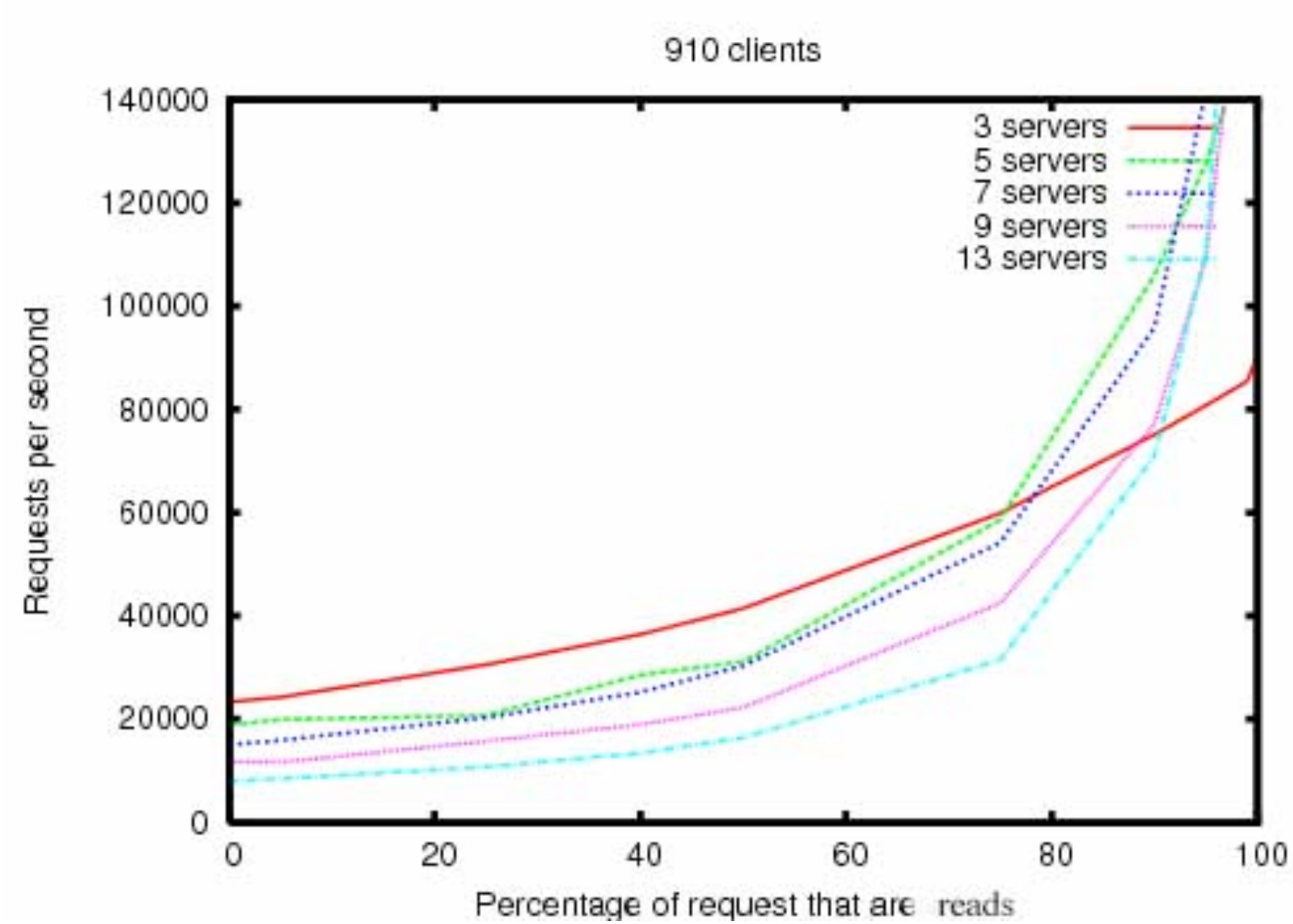
Metrics

- Functionality
  - Very basic functionality; powerful abstraction
  - Not meant to be used as a database
    - Recipes for barriers, queues, locks, 2PC, leader election
- Portability

Metrics

- Ease of use
  - Easy to set up
  - ZK client supports Java and C
- Maturity/community
  - Paper published in 2010
  - 1443 commits
  - 1464 questions on StackOverflow

Performance



https://zookeeper.apache.org/doc/trunk/zookeeperOver.html

Performance

- Latency
  - 130 - 140 ms for create & set
  - 19 - 26 ms for get

Redis

System overview

- In-memory key-value store (“data structure store”)
  - key: binary-compatible
  - value: supports many data structures
    - Strings, lists, sets, hashes, sorted sets, bitmaps, etc
- Transactions
  - Serializability
  - Atomicity
  - No rollbacks

System overview

- Data sharding
  - key mapped to hash slot
  - each node responsible for a subset of slots
- Persistence
  - database snapshot
  - write logging (different rates of fsync)
- Replication: master-slave; async

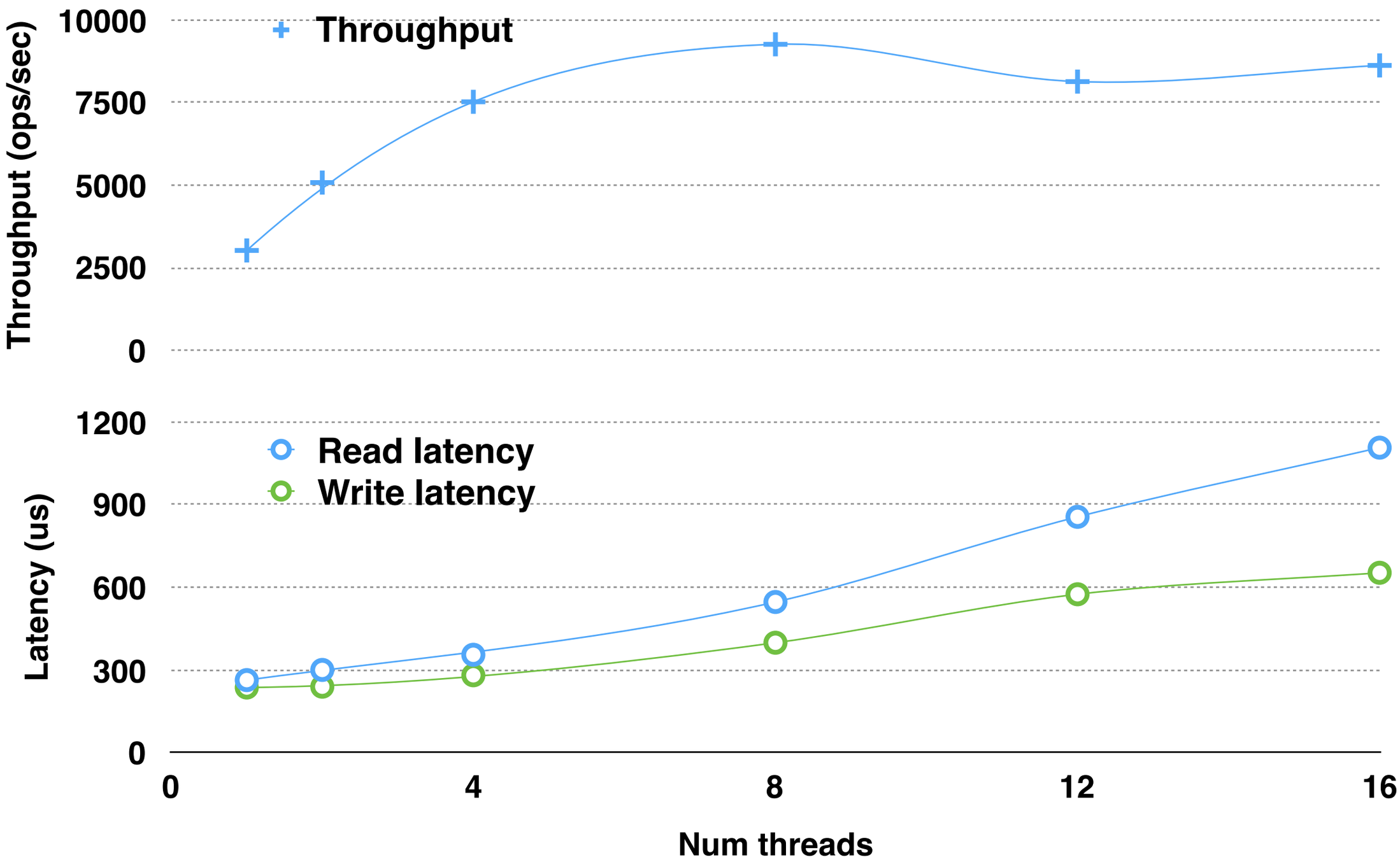
Metrics

- Consistency
  - Not able to guarantee strong consistency
    - Possible to lose writes that were acknowledged to the client
    - Asynchronous replication
    - Can provide synchronous replication using WAIT

Metrics

- Functionality
  - Many data structures supported for values
- Portability
- Ease of use
  - Easy set up
- Maturity/community
  - 5857 commits
  - 9354 StackOverflow questions

Performance (YCSB-a)



Redis - pub/sub

System overview

- In-memory key-value (data structure) store that has many utilities
- Can be used as a publish/subscribe system
- Clients can use the key-value store to communicate
- Publishers write to channels
- Subscribers receive messages from subscribed channels

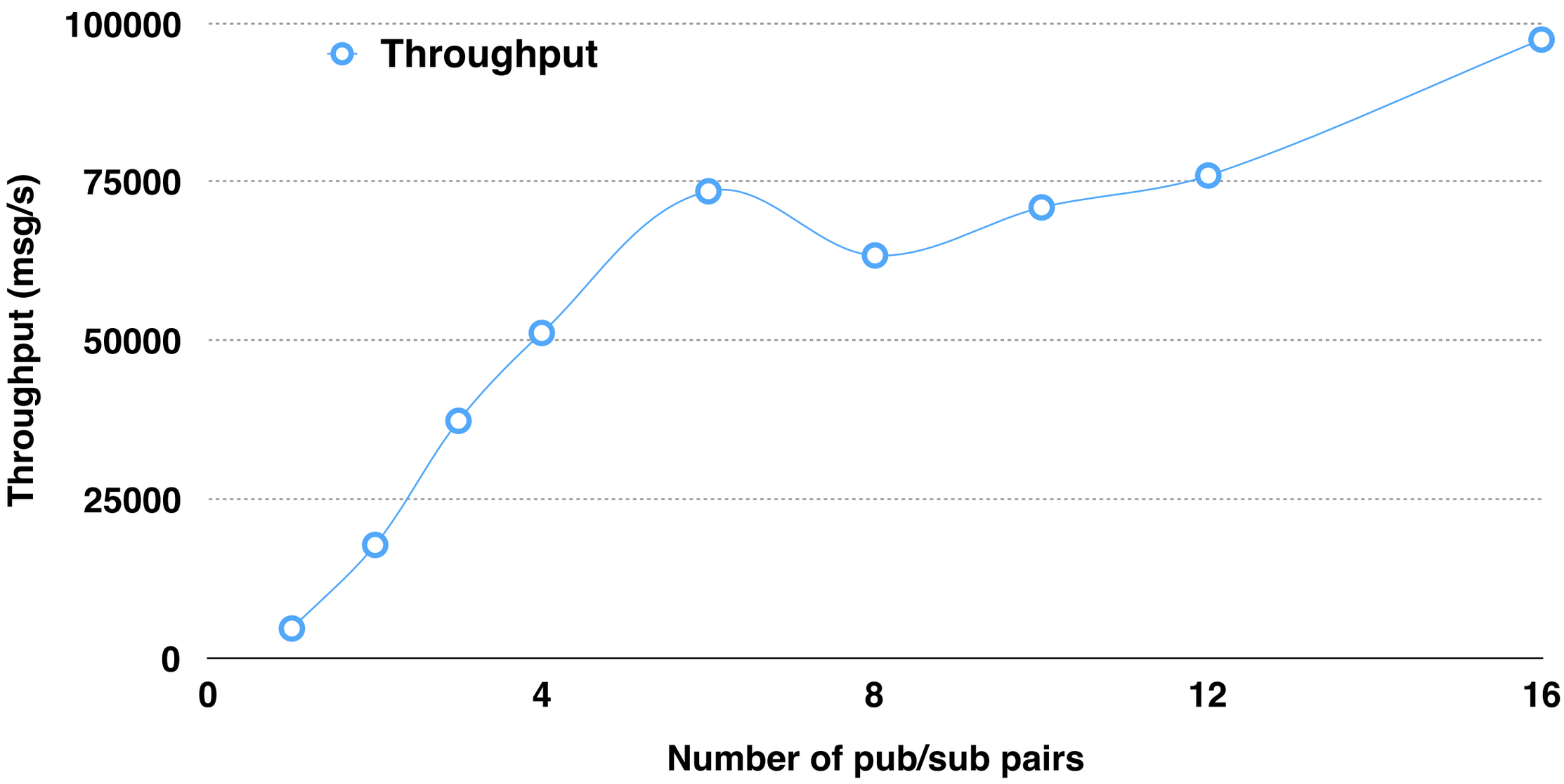
System overview

- PUBLISH
- SUBSCRIBE
- UNSUBSCRIBE
- PSUBSCRIBE
- PUNSUBSCRIBE

Metrics

- Reliability
- Functionality
  - pattern matching support
- Portability
  - many languages supported
- Easy to set up (cluster mode needs more)
- Maturity/stability
  - 5857 commits
  - 9354 StackOverflow questions

Performance



Latency: 0.18 ms