# Memcached vs Redis

Benchmarking in-memory object caches

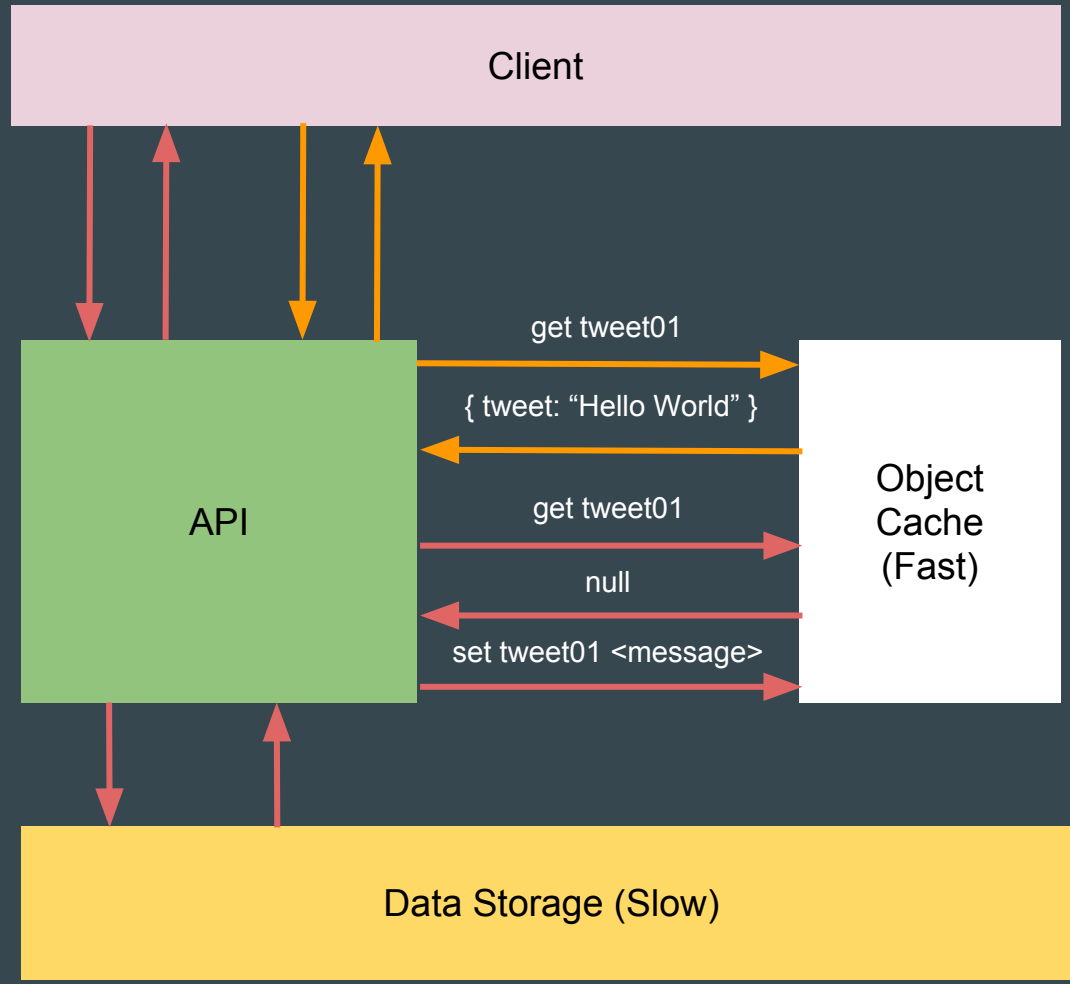
Milan Pavlik

# Object Caches

# Object Caches

- Store results of expensive operations
- Key-value
- Arbitrary values

- Basic operations:
  - get
  - set
  - delete



# Memcached

- Distributed
- Multi-threaded
- Requires locking
- In memory only
- Simple API
  - get
  - set
  - mget

# Redis

- Single-threaded
- Atomic operations
- Can write to disk
- Rich API
  - get, set, mget
  - Sets
  - SortedSets
  - Arrays
  - HashTables
  - HyperLogLogs (size estimation)

# Memcached

- Distributed
- Multi-threaded
- Requires locking
- In memory only
- Simple API
  - get
  - set
  - mget

# Redis

- Single-threaded
- Atomic operations
- Can write to disk
- Rich API
  - get, set, mget
  - Sets
  - SortedSets
  - Arrays
  - HashTables
  - HyperLogLogs (size estimation)

**How do they compare on a normalized feature set?**

# Benchmarking setup

- 8 machines (quad-core,  8 GB Memory)
  - 1 server
  - 7 clients
- Single rack (1 Gbps link)
- Clients generate traffic
- Server handles responses
- Monitor latency distribution of responses

# Data

- Simulate real use cases

- Twitter - Tweets
- Wikipedia - Articles
- Flickr/Instagram - Images & Image thumbnails

# Timeline

- S1 Weeks 1 - 4: Infrastructure & Benchmarking tools research
- S1 Week 5: Infrastructure tools - automation of data collection and processing
- S1 Weeks 6 - 7: Performance testing on basic variables of the setup
- S1 Weeks 8 - 10: Iterate on performance tests given previous results
- S1 Weeks 11 - S2 Week 0: Analysis and/or further performance tests

- S2 Week 1 - 2: Analysis & Writing Interim report
- S2 Week 3+: Iterate on Analysis and Write the report

# Q&A