



REDIS

DATABASE CACHING



What is REDIS ?

- Data Structure Store
 - used as database
 - used as message broker
 - used as **cache**
- Work in RAM, dump on disk for persistence
- A server
 - network capability



Who's using it ?

- Twitter
- GitHub
- Pinterest , Snapchat
- Craigslist
- Digg, *StackOverflow*, Flickr



What can it do ?

- **Transactions**
 - Group commands, execute them at once, atomic
- **Pub/Sub message paradigm**
- **Key Time-to-live**
- **LRU replacement algorithm**
 - LFU (redis 4.0)
- **Monitoring, notifications, scaling**
 - Redis Sentinel, Cluster



Redis Sentinel

- **Monitoring**
 - Constantly check M/S status
- **Notifying**
 - Tell admin about system state
- **Automatic Failover**
 - Slave k become Master
 - Slaves reconfigured
- **Routing provider**
 - Client connect to it to get adress of master with data needed



Redis Cluster

- **High performance and linear scalability**
 - Up to 1000 nodes
- **Acceptable degree of write safety**
- **Availability**
 - Survive partitions
 - Replicas migrations



Functional Advantages

- **Really fast**
- **Supports many data types**
 - String, Lists, (Sorted) sets, ..
- **Easy partitioning**
- **Automatic Failover**
 - Sentinel



Functional Disadvantages

- **Not ACID (atomicity, consistency, ..)**
- **Master/Slave replication architecture**
 - More load on node master
 - Single point of failure (master)
 - Data can be lost
- **Represented dataset need to fit in RAM**
- **Hard to do request on all dataset**



Non-Functional Advantages

- **Open Source**
 - Vibrant community
- **Easy to install**
- **No Dependencies**
- **Automatic Failover**
 - Sentinel
- **Client API in every popular language**



Non-Functional Disadvantages

- **No access control**
- **Harder to scale**
 - Redis Cluster
 - Heroku
- **Memory fragmentation issues**



Comparison with Memcached

- **Read/Write speed**
 - Similar, fast
- **Memory usage**
 - Redis give back unused memory, not memcached
- **Persistence**
 - Redis does I/O dumping by default, not memcached
- **Scaling**
 - Redis Cluster, nothing for memcached



Conclusion

- **Hugely used**
 - Best when used as cache with other DB
- **Scalable**
 - Used by big services
- **Flash-like Fast**
- **Available as service**
 - AWP offer his own redis service