



Redis

In C#



What is Redis?

- Key-Value Store
- Memory-based
 - which means...
 - very fast read and write!!!
- But with periodically flushing to disk, so you enjoy all the goods.
- Simplicity

Who uses Redis?





Let's Begin



Begin Redis

- String
 - The simplest data type in Redis
 - It can be anything.



String

- Binary safe
- And ideal for:
 - Plain strings
 - Full JSON objects
- Commands
 - SET [*object-key*] [*value*]
 - GET [*object-key*]
- With expiry.
 - SET [*object-key*] [*value*] [*expiry*]



Other Types

- List
- Hash
- Set



Use Cases



1. Cache

- A degree of data staleness is acceptable
- Some data doesn't need a full-fledged RDBMS



2. Session

- Session data can be stored outside the web server machine
- We can deploy/remove machines without affecting session data currently being used



Demo



How Jetabroad Use?

- Move search results originally stored in web server machine memory to Redis.



Settings

- Load Model:
 - 150 requests/minute -> 2.5 requests/seconds
 - 4 Mb/request (need to be cached)
- Cache Settings: 30 minutes expiry
- Original Search Response Time: 11 ~ 13 seconds



Results

- Resource Consumption in the Redis instance:
 - CPU: ~4%
 - RAM: ~18 Gb
- Search Response Time
 - insignificantly increased



Think about it...

- Type of usage. Persistent/Temporary data store or Cache?
- Data Capacity Plan



Think about it...

- Eviction Policy

- Least Recently Used (LRU)
- Redis provides flexibility on eviction policy to use.
- “noeviction”, “allkeys-lru”, “volatile-lru”, “allkeys-random”, “volatile-random”, “volatile-ttl”
- <http://redis.io/topics/lru-cache>



Questions?
