Using Redis in Game System

By Triton Ho@Mars' Potato

Game System Challenges

- Short request handling time
- Massive data change
- Low data lifespan
- Naturally high concurrency contention

Luckily

- Game data is usually fault-tolerance
  - Users will f**k the developers and keep playing
- Usually, game data is not large

Key Concept of Redis

- a noSQL database
- In-memory, but persistent
- Key-value, with various data structure
- Allow Time-to-Live
- Allow blocking IO

What Redis can do?

- Store game lobby data
- Session database
- High performance lock server
- Simple but enough message

Disadvantages

- Huge increase in development mandays.
  - For first project, ~50% extra mandays in server side
- One extra component in server side, extra effort on monitering and failover
- Need explicit lock.
- Must prepared for the inconsistent state after crash-and-recovery

But, why still Redis?

- Memory is cheap and WILL BE cheaper
- For high traffic game server, Redis is still easier than a group of RDBMS server.
- In VM instance, disk IO is usually shared, but memory bankwidth is not.