Faster and higher availability than WAG
(WOS Access Gateway)

ObjectFS: Native File System on WOS

Overview: CLIENT Solution vs SERVER Solution in WAG

- Hardware Specs： No special requirement on clients
- Software Components
  - ObjectFS – FUSE (On Linux)/DOKAN (on Windows) based file system
  - Cassandra for storing metadata

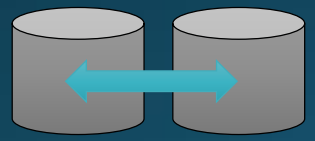
Architecture: Native and Direct vs Complex Protocols in WAG

Command –Line Utilities

ObjectFS for Linux and Windows
FUSE/DOKAN-based file system

Cassandra API
Metadata

File Cache
file system



RAID1 – SSD
Used for OS and Cassandra metadata tables

NFS
Ganesha

CIFS
Likewise

wactl
Command-Line Utility

Web UI

Corosync/Pacemaker
Resource Management and High Availability

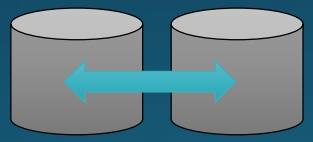
wosfs
FUSE-based, file system

RabbitMQ
Namespace Synchronization

CentOS 6

Postgres
Metadata

Staging
ext4



RAID1 – SSD
Used for OS and Postgres metadata tables

RAID6, 4+2 – 15K SAS
Used for data cache, namespace sync logs and other protected data

ObjectFS: Direct WOS Access

- WAG has a CENTRAL Staging Area.
- Except file modifying, ObjectFS access WOS directly without delay and discount.
- File Cache: Only when modifying
  - Why?
    - File system write protocol and WOS object protocol not compatible
    - File modification is not supported using object protocol
  - Distributed
    - Every client can have its own cache.

Basic File System Functions at API Level

- Basic Read/Write/List/Rename/Create/Truncate Operations
- Further operations such as Hard Link/ Symbolic (soft) Link, Chmod, Chown are not supported so far.
- Also, any objects can be accessed as file by its OID (as file name) with 100% performance by pending patent of ZeroFS®
  - Such as, ~/ofs/FAExQJWyBmcelRARAsxZmKK56AJXjFXCHqgSCT6I.mkv

Extra File Ingest Utilities

- to ingest files to WOS quickly
- An example has been open source at https://github.com/MRYingLEE/ObjectFileSys
  - Quicker: Multi-threaded
  - Flexible: open sourced

The next step

- To provide a Windows platform Edition (Only when clients required)
- To improve Performance and Robust (Long term work)
  - Meta access has 10x times efficiency potential by meta cache.
  - File access has 2-5x times efficiency potential by file cache and multi-threaded coding.

Thank you!

- MR.YING.LEE@gmail.com
- WeChat: ying713560
- Mobile/Whatsapp: +852-98178373