Cost-effective storage Services with Scality RING
Status 2012

- Traditional Fileservices -> Expensive
- Economical Fileservices -> Silo
Use Cases

• File-service backend:
  – Scalable
  – Economical
  – Robust
• OwnCloud
• iSCSI
• Backup Disk-Pool
Why Scality 1

• Robust and performant FUSE-Connector -> “Scale Out File System”
  – Virtual POSIX filesystem
  – Max. $10^{20}$ Files/mountpoint (namespace)
  – No filesize limit

• Proven and performant erasure coding implementation (ARC)
Why Scality 2

• All cluster-nodes are (really) equal
  – Shared nothing
    • SSD failure does NOT take data disks offline
      – BUT...
    • No monitor role
  – Fully distributed data and metadata
• 160-bit Keypspace
  – Good expandability
    • Not so sure about SWIFT
Basic Object Access
SOFS Access

Ring access through the SOFS (Scale Out File System) connectors

Diagram explaining the connections and processes involved in SOFS access.
Connectors

- FTP CLT
- CIFS CLT
- AFP CLT
- NFS CLT

- FTP SRV
- CIFS SRV
- AFP SRV
- NFS

- FUSE
- FUSE
- FUSE
- CDMI

- HTTP PUT/GET
- HTTP PUT/GET

- MD RING

DATA RING
What does it look like?
What about Hardware?

- 16 Servers
- 2x System HDD
- 2x 400GB SSDs/server
- 32x 4TB HDD/server
  - 2 PB raw capacity
- Dual Port 10Gb Ethernet
Experiences

• SOFS Reliable
  – (Current version: VERY reliable)
• Very good throughput with ARC
  – Up to 500MB/s (backend)
• iSCSI works surprisingly well
• Excellent support
Questions?