Electronics and Telecommunications Research Institute
ETRI 한국전자통신연구원

GLORY-FS

- 저비용 대규모 서버 기반 글로벌 파일시스템 기술 -

※ GLORY: Global Resource Management System For Future Internet Service

2008. 4. 25

김영균
(kimyoung@etri.re.kr)

저장시스템연구팀/인터넷플랫폼연구부

목차

Electronics and Telecommunications Research Institute
ETRI 한국전자통신연구원

GLORY 프로젝트 소개

Fund: Ministry of Knowledge and Economics, 2007 ~ 2012(5 years)

Storage System Research Team

Electronics and Telecommunications Research Institute
ETRI 한국전자통신연구원

인터넷 서비스 솔루션 진화

국내외 산업체 기술 동향

Google

- 강력한 웹 검색엔진
- 저가의 고성능 서버팜
- 파괴적 수익모델(무료)

➔저가의 서버 플랫폼 기술로 전세계 서비스 시장 선점 시도

MS

- MS Live 서비스
  : 웹을 통해 SW와 저장 공간을 서비스화

➔시장 수성 및 구글 견제

EU, JAPAN

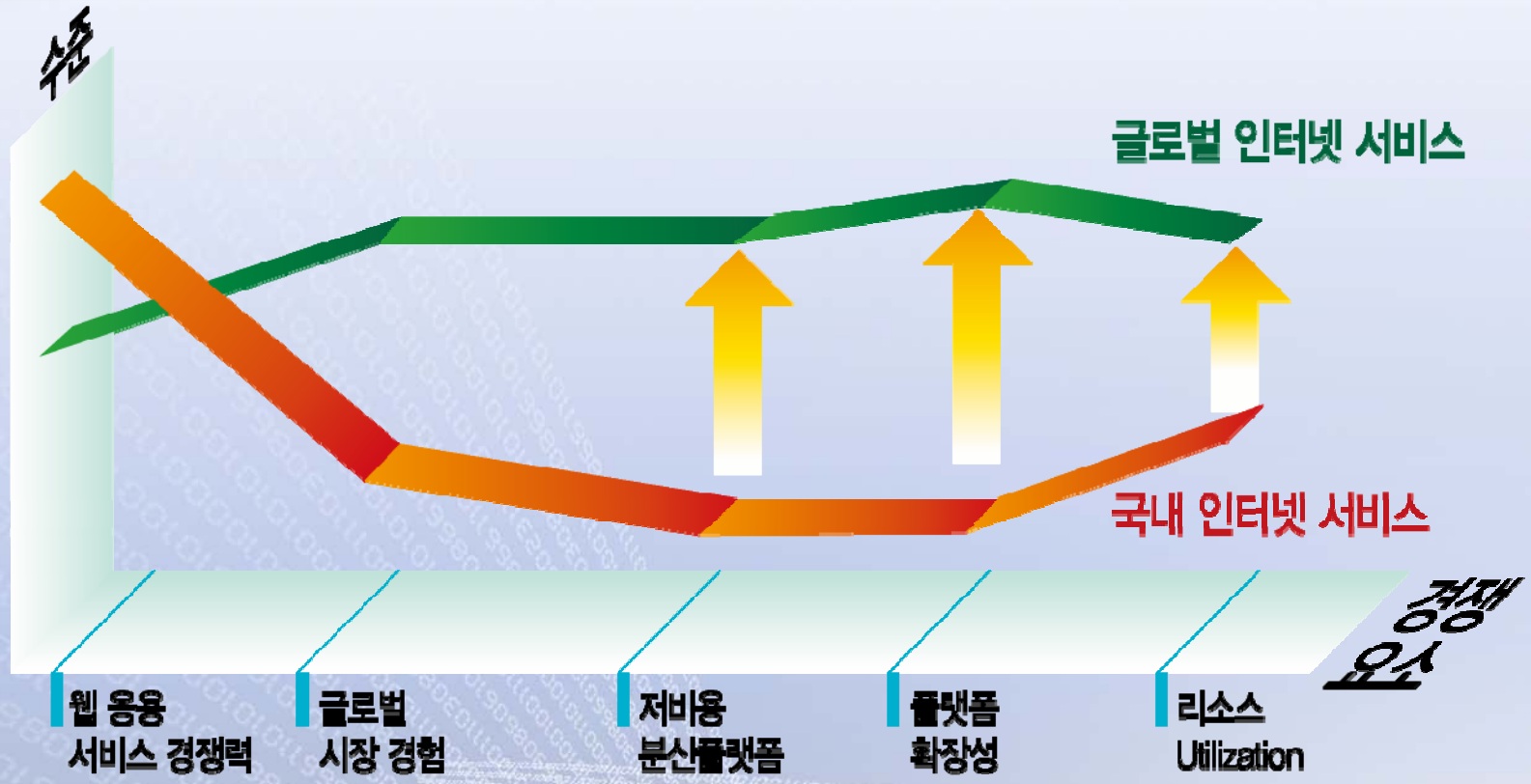
- **일본, 독일, 프랑스**
  산학연 협동 멀티미디어 검색 기술 개발 착수
- **일본 : 대항해 프로젝트**
  38개기관 수입억 엔/년 투자 3-5년 후 사용화
- **EU**
  Quaero, 9000만 유로/년

국내 업계 동향

- **국내 인터넷 인프라는 세계적 테스트베드** : Nate 모바일 인터넷, cyworld, 다음 카페, Naver 지식검색
- **플랫폼 기술 한계로 확장성 제한** : naver.com, daum.net, nate.com, paran.com, empas.com

미래 인터넷 서비스 산업을 강화하기 위한
저비용 대규모 글로벌 인터넷 서비스 솔루션

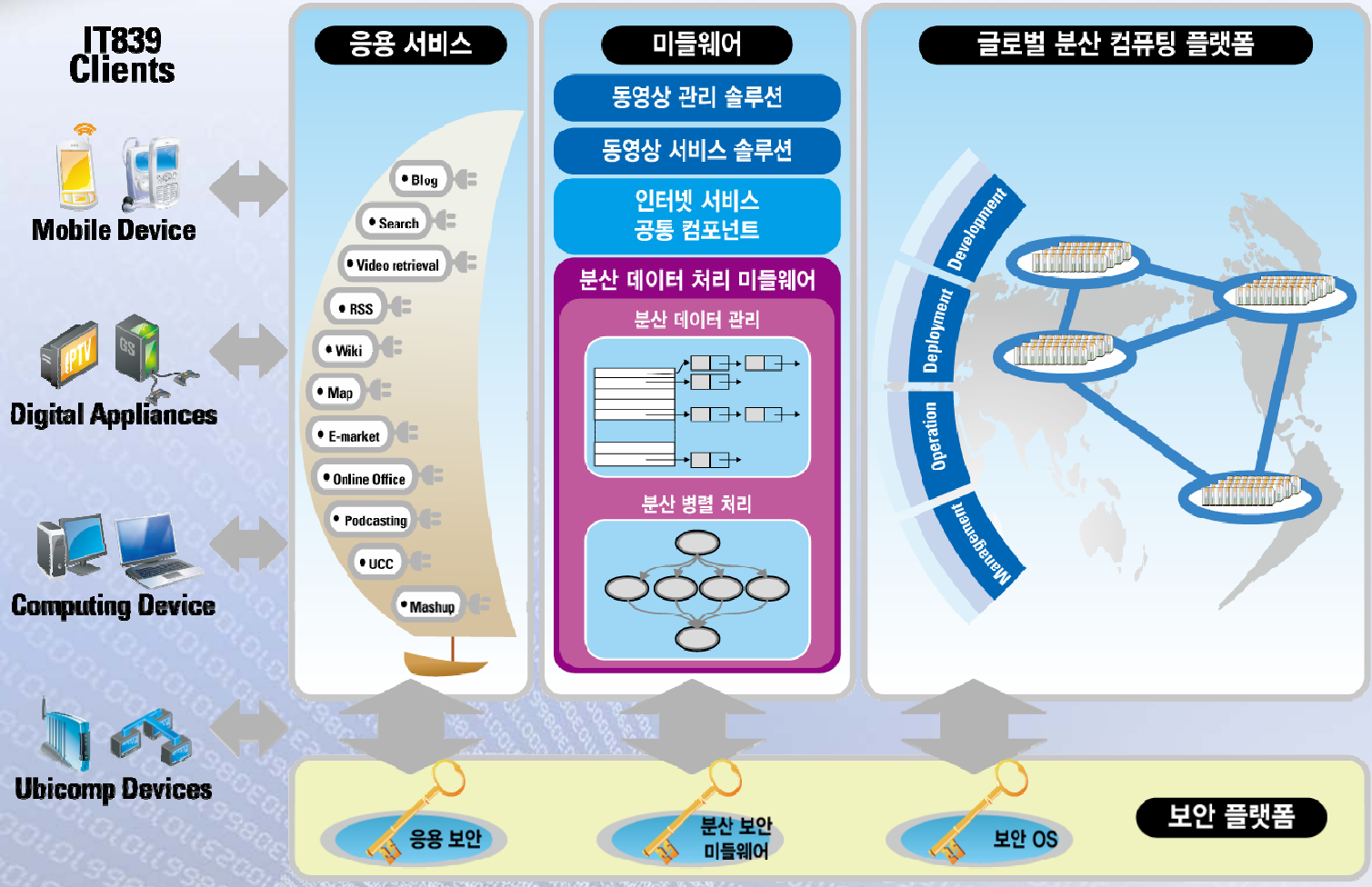
국내 인터넷 서비스 기업 경쟁력



Killer App.: 동영상 서비스

- **UCC · 동영상 등 다양한 콘텐츠의 증가로 검색 지평도 확대**
  - IBM, Blinkx, TVEyes 등
- **UCC 기반 새로운 동영상 수익 모델**
  - **광고삽입**: **곰**TV(**그라텍**), **판도라티비**, **아프리카**(**나우콤**) 등
  - **수익배분**: **구글**, **레버**(Revver) 등
  - **동영상 기반 오픈 마켓**: Brightcove(**미국**)
- **인터넷 동영상을 유통시키는 신디케이션 모델**
  - Maven Networks(**미국**)

저비용 대규모 글로벌 인터넷 서비스 개념도



GLORY 최종 목표

UCC, IPTV 등 동영상 기반 新 인터넷 서비스 개발 및
저비용 대규모 글로벌 분산 컴퓨팅 플랫폼을 공개 S/W 기반으로 개발하여
글로벌 인터넷 서비스 토털 솔루션을 제공

- 동영상 내용기반 검색 서비스 구축 지원
- 엑사바이트 규모 콘텐츠 저장 공간 제공
- 대규모 인터넷 서비스를 위한 백만 노드 확장성 제공
- 인터넷 서비스 가용성 99.999% 지원
- 서버 고효율 운영으로 노드 수 절반 감축

GLORY 솔루션 구성


동영상 기반 인터넷 응용 서비스
UCC 검색 서비스
IPTV 서비스
웹 문서 검색 서비스
인터넷 서비스 공동 컴포넌트
동영상 관리 솔루션
제 작
태 깅
저 장
검 색
동영상 서비스 솔루션
동영상 분배
동영상 스트리밍
분산 데이터 처리 미들웨어
대규모 분산 병렬처리
작업분해 및 병합
분산작업 스케줄링
대용량 데이터 분산관리
서비스 데이터 관리
데이터 분산 배치
데이터 접근 및 복구
분산 보안 미들웨어
사용자/ 서비스 인증
장치/ 커널 인증
글로벌 클러스터 관리 솔루션
클러스터 구성 및 자원관리
자동 프로비저닝
분산자원 모니터링
글로벌 파일 시스템
파일 메타데이터 분산 관리
파일 분산 저장 및 복제
파일 원거리 백업/보존
플랫폼 운영체제 및 하드웨어
저전력 운영체제
ACTIVE BIOS/노드 관리자
저전력 하드웨어

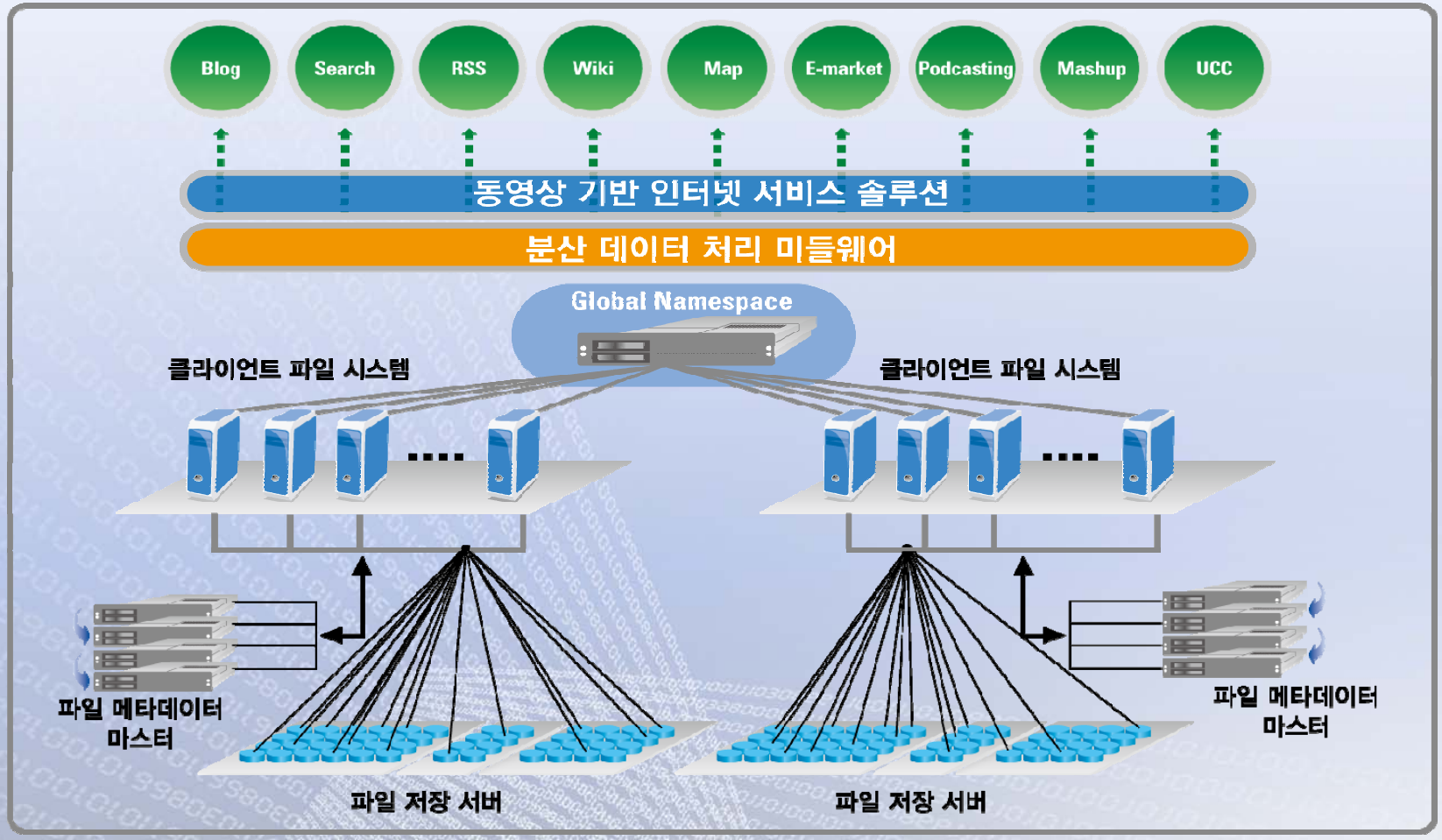
Electronics and Telecommunications Research Institute
ETRI 한국전자통신연구원

GLORY-FS

LakeFS: Towards highly manageable cluster storage for extremely scalable services
- to be published in DS2 Workshop, Italy, June, 2008

Storage System Research Team

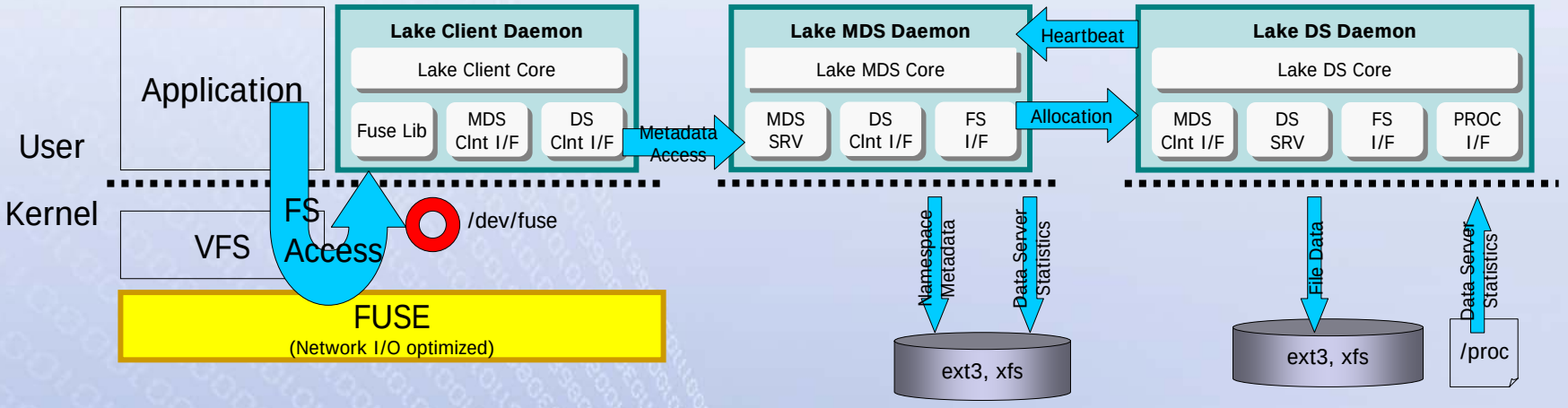
LakeFS Big Picture



LakeFS: Goals

- **Highly scalability for both storage space & storage bandwidth**
  - up to 10K clients: more than 150GB/s
  - physically, 10K data nodes: more than 10PB capacity storage
- **Fault-tolerance of storage: 99.999% availability**
  - file system metadata availability (metadata cluster)
  - file data availability (replication)
- **Intelligent storage administration: Self-***
  - configuration, monitor, healing, etc.

LakeFS: Architecture



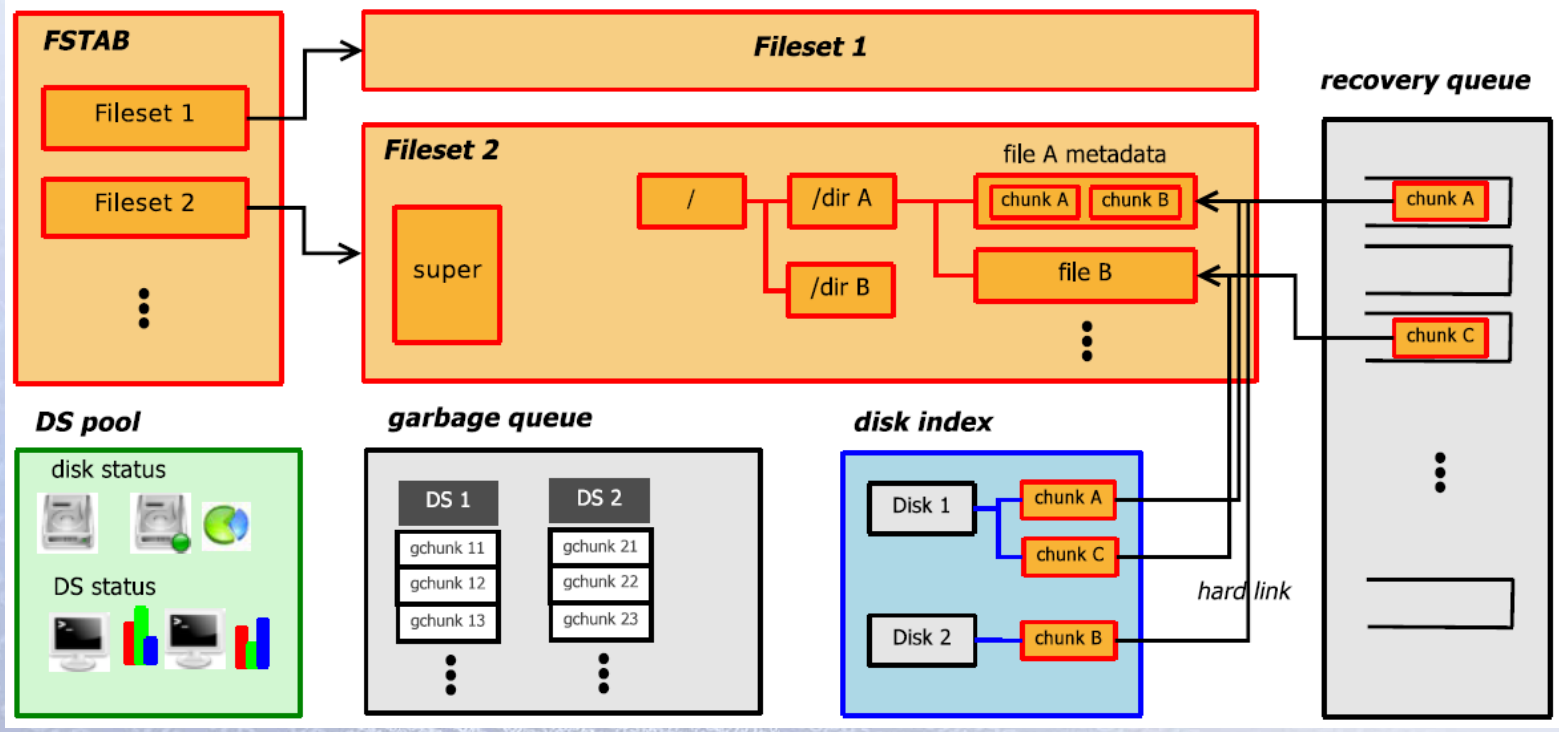
LakeFS: Client

- **POSIX-compliant API support**
  - fcntl, lockf (△)
  - flock (x)
  - mmap (writable shared mmap, o), open (O_DIRECT, x)
- **Cache consistency**
  - session-semantics consistency model (lock-free coherency control)
  - focusing on large, sequential read intensive workloads

LakeFS: Metadata Server (1)

Metadata storage

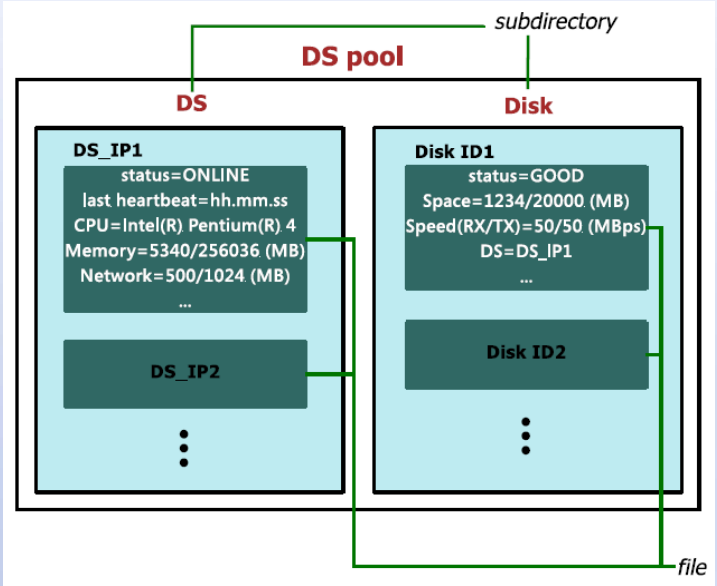
- namespace tree/DS pool/disk index/recovery queue/garbage queue



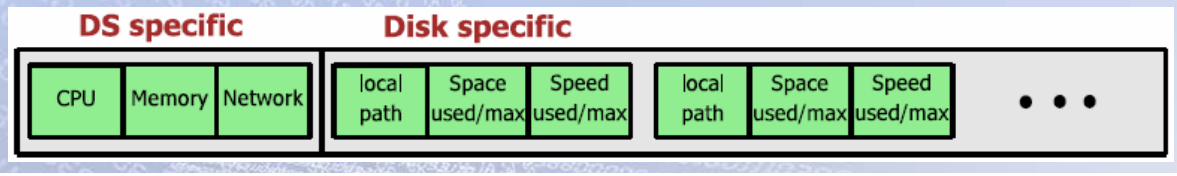
LakeFS: Metadata Server (2)

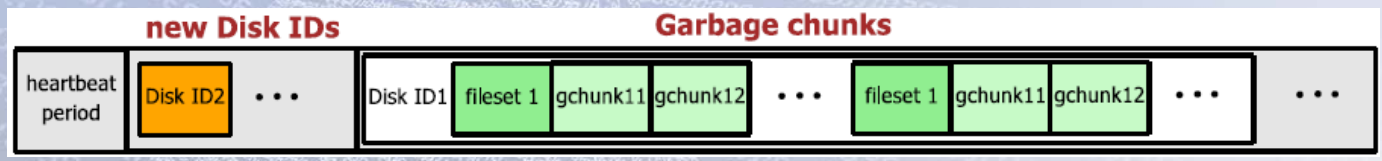
DS management

- DS status: disk space utilization,
  io throughput,
  cpu/network/memory utilization



- heartbeat messaging



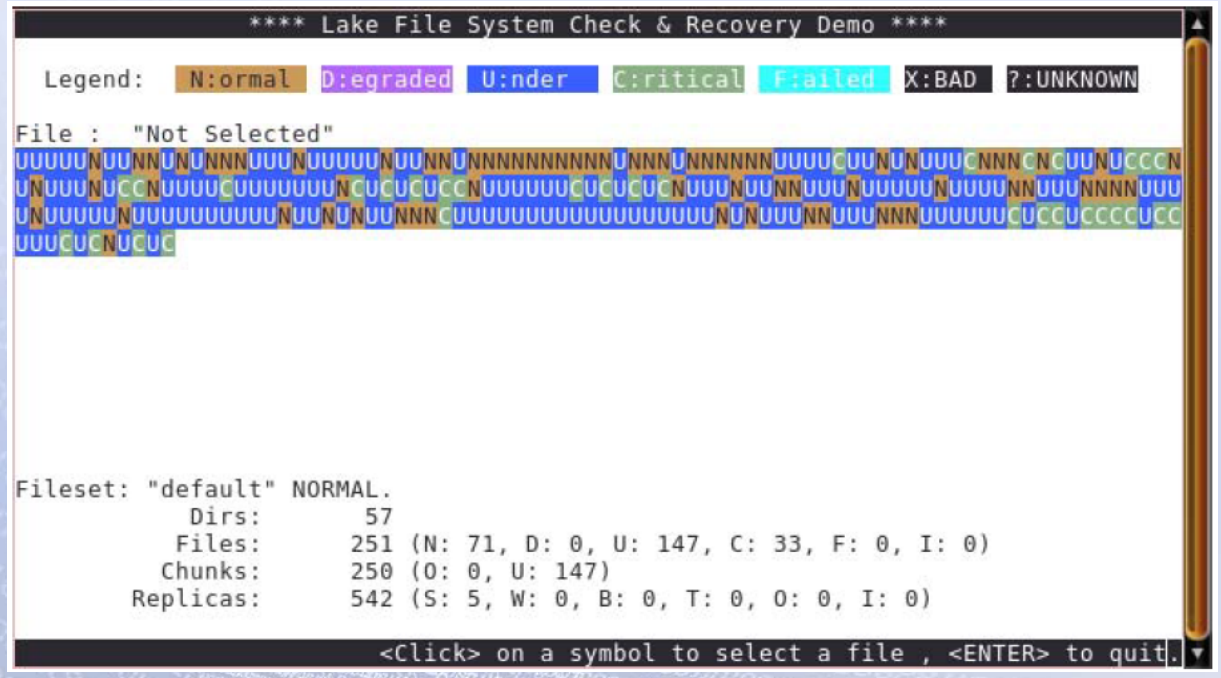


LakeFS: Metadata Server (3)

File replication

- "**declustered chunk replication**"

  : files are striped across component chunks on different DS

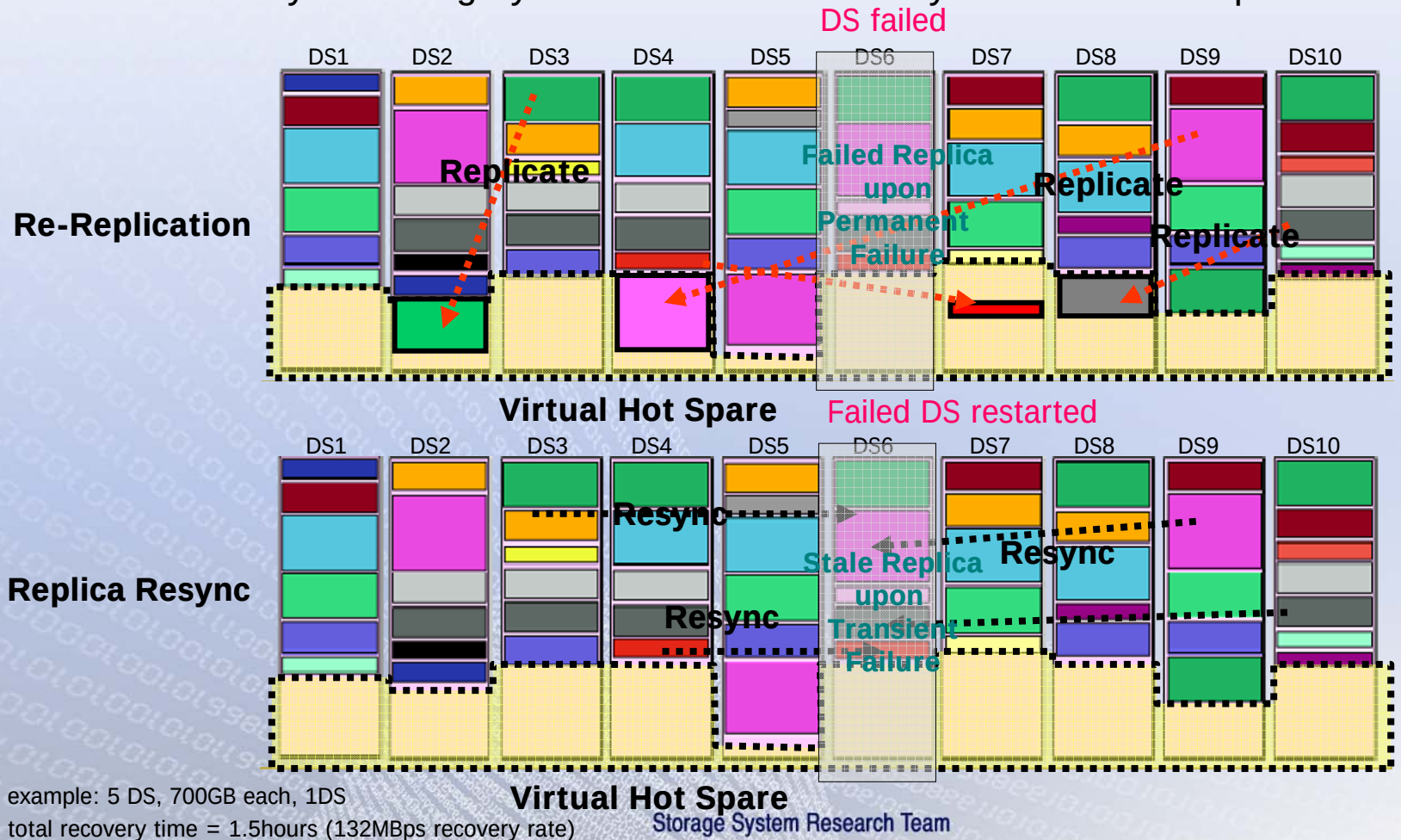


- "**real-time & lazy replication modes**"
- "**intelligent chunk placement policy: NUFA/RR/UC/UL**"

LakeFS: Metadata Server (4)

Recovery

- file system integrity check: check and classify each file status upon DS failure

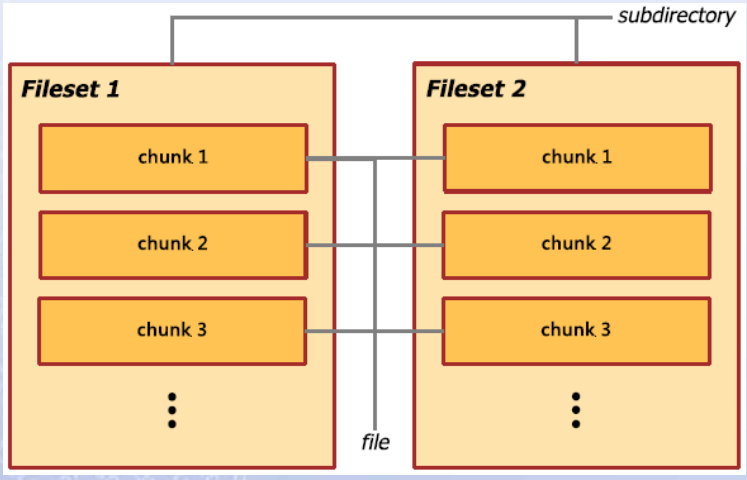


example: 5 DS, 700GB each, 1DS
total recovery time = 1.5hours (132MBps recovery rate)

LakeFS: Data Server

Data storage & management

- store multiple fixed sized chunks into regular files of the local file system



- version management for chunk integrity

Parallel chunk replication

LakeFS: Self-* Utilities

Monitoring

- DS & MDS status information

```
Data Server    RACK CPU%   Mem  Free  Net     TX     RX Status
129.254.202.141   1  11  1009M   14M   1G  366K   13M [ONLINE]
        DiskID   Total  Used  Free  Read Write Status
        0e901368   66G    4G   62G 3243K   14M  [GOOD]
        Total      66G    4G   62G 3243K   14M
129.254.202.143   1  65  1009M   18M   1G   13M   19M [ONLINE]
        DiskID   Total  Used  Free  Read Write Status
        1eec624a   66G    4G   62G 6014K   12M  [GOOD]
        Total      66G    4G   62G 6014K   12M
129.254.202.146   1  41  1009M   13M   1G    4M   22M [ONLINE]
        DiskID   Total  Used  Free  Read Write Status
        2c7e17b2   66G    4G   62G 4463K   19M  [GOOD]
        Total      66G    4G   62G 4463K   19M
129.254.202.148   1  21  1009M   14M   1G    6M   17M [ONLINE]
        DiskID   Total  Used  Free  Read Write Status
        169fd4b3   66G    4G   62G 6541K   18M  [GOOD]
        Total      66G    4G   62G 6541K   18M
129.254.202.152   1  38  1009M   14M   1G    1M   25M [ONLINE]
        DiskID   Total  Used  Free  Read Write Status
        521fb389   66G    4G   62G 2336K   22M  [GOOD]
        Total      66G    4G   62G 2336K   22M
[root@fm4 bin]#
```

(a) DS and disk status

```
[root@fm4 bin]# ./lfs_lsmds
  Operation          ops ops/sec
     GETATTR:       3114      40
     SETATTR:        308       5
       CREAT:        621       8
       MKNOD:          0       0
       MKDIR:         56       0
     SYMLINK:          0       0
        LINK:          0       0
      UNLINK:        188       3
       RMDIR:          0       0
      RENAME:        336       5
    READLINK:          0       0
     READDIR:          0       0
     GETXATTR:      1447      20
     SETXATTR:      4805      25
   LISTXATTR:          0       0
REMOVEXATTR:           0       0
  ALLOCCHUNK:        654       8
    TRUNCATE:          0       0
      STATFS:          0       0
GETFSETATTR:           8       0
SETFSETATTR:           0       0
       TOTAL:      11537     117
```

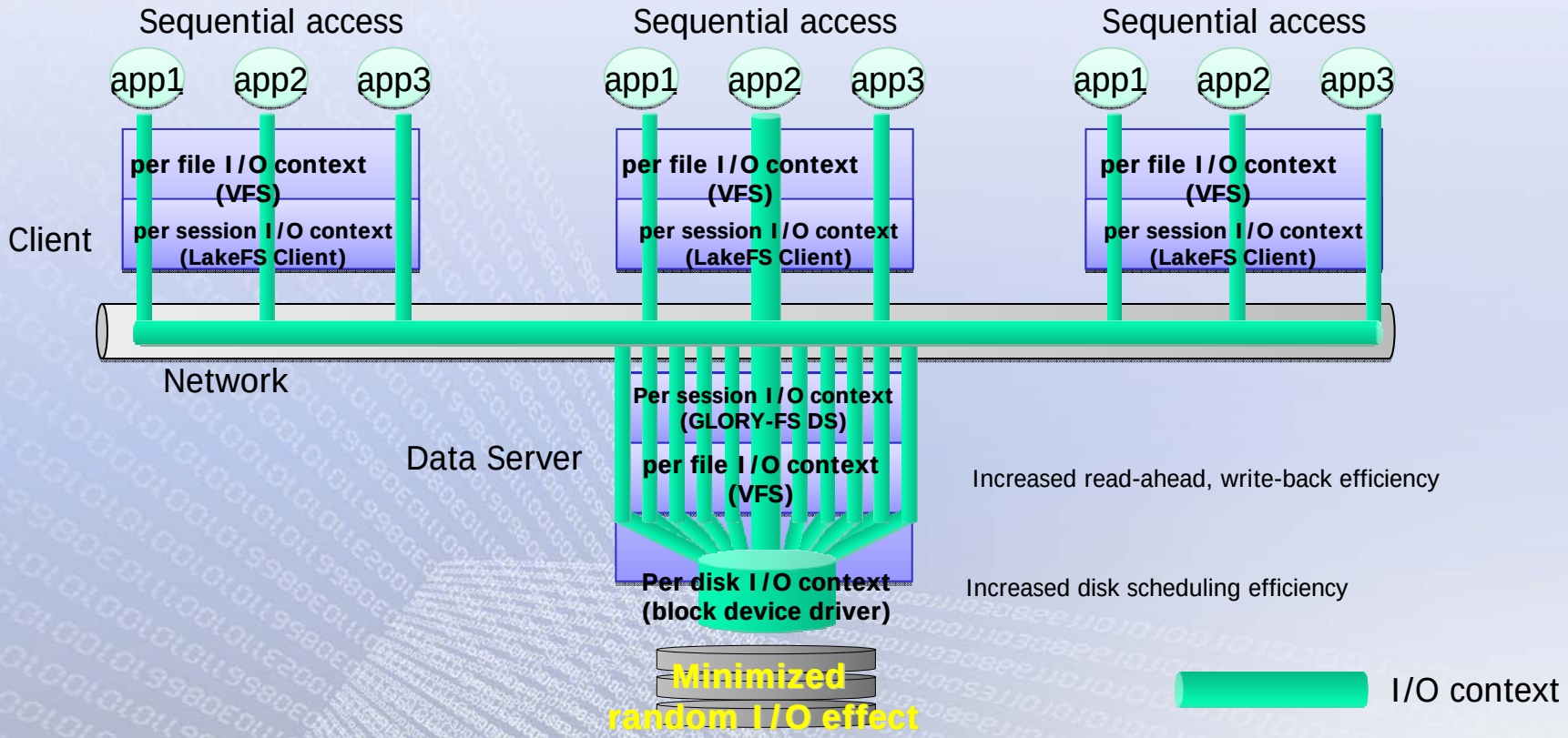
Customizable event handler

- the first step to intelligent administration

LakeFS: I/O Optimization

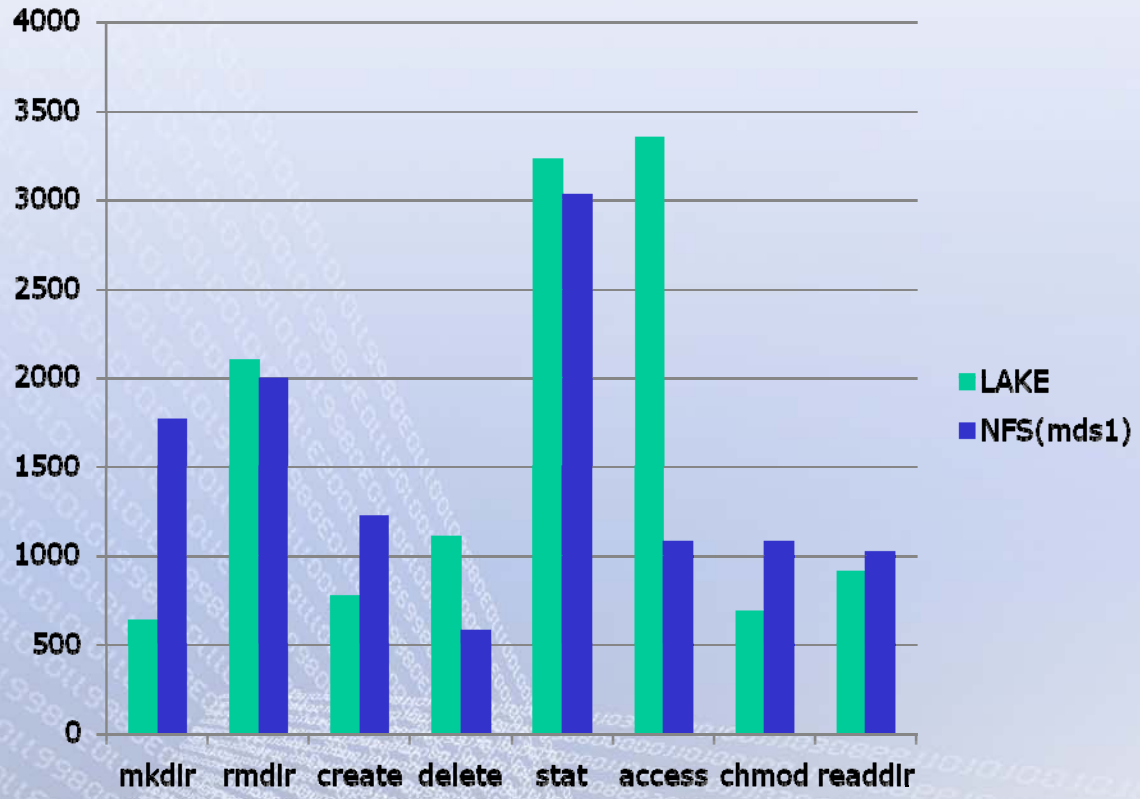
Pipeline effect while I/O

- optimization for large sequential read intensive workloads



LakeFS: Performance (1)

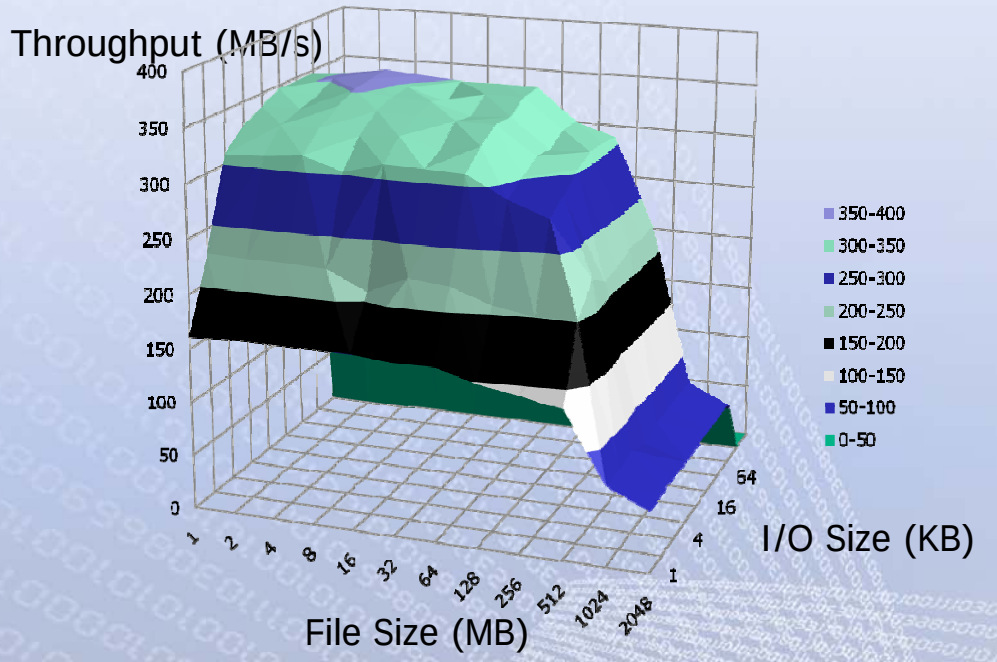
Metadata operation performance



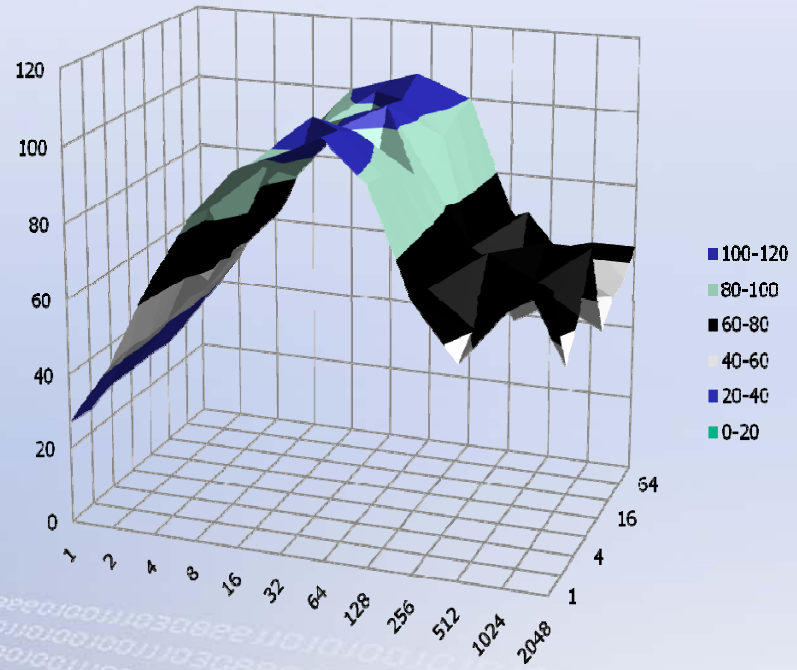
LakeFS: Performance (2)

- **Read / write performance**
  - single client & single DS

write operation

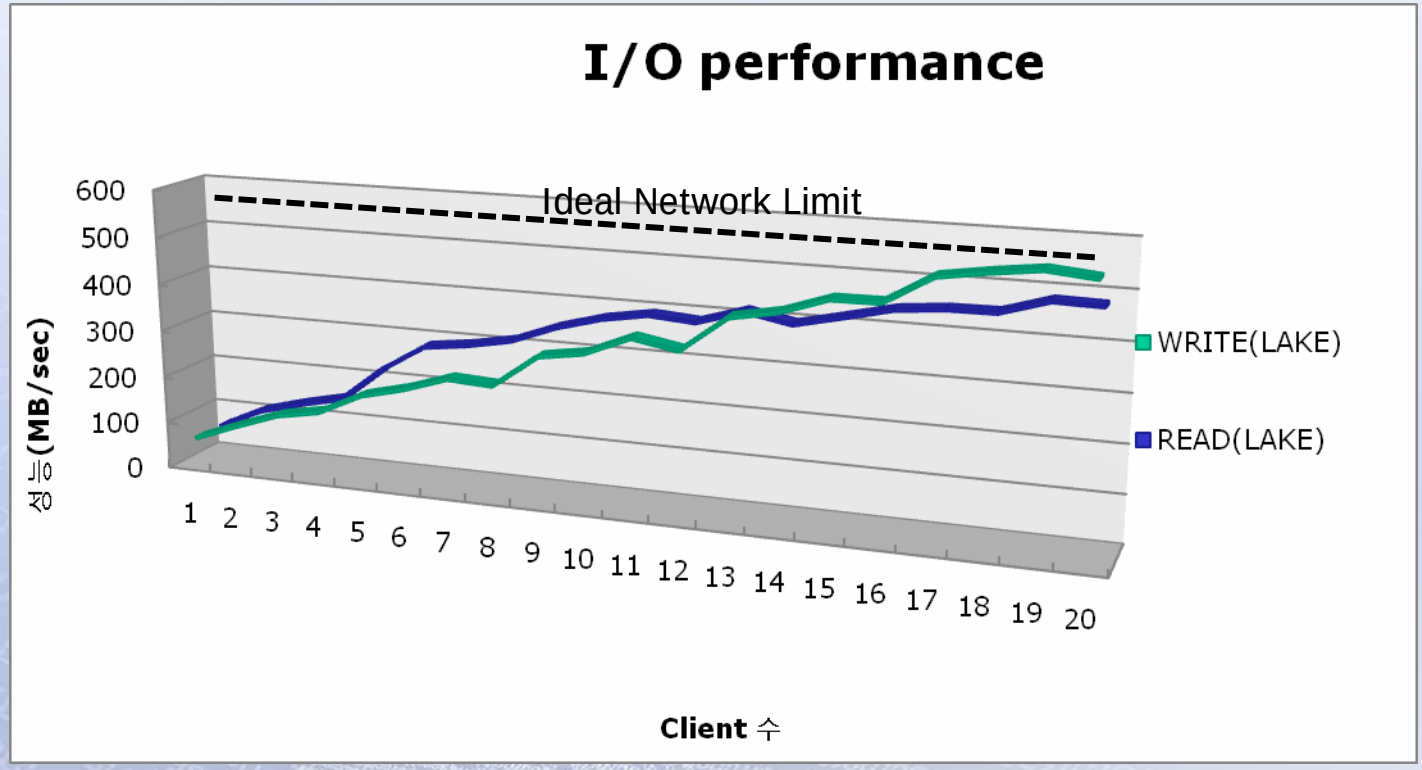


read operation



LakeFS: Performance (3)

Aggregate IO performance



LakeFS: SSD Deployment

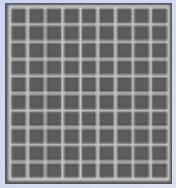
Metadata repository

- try to increase metadata operation performance
- simulation
  : Seagate Barracuda 7200(320G) vs. Mtron MSD-SATA6025(SATA, 32G, 2.5”)
  : file size(250 Bytes), # of files : 8 million files

Operation	HDD	SSD
Create/Write	2715	2631
Sequential Read	6250	6027
Random Read (1 Thread)	56	2105
Random Read (20 Thread)	115	5714

# of Thread	HDD	SSD	ratio
1	56	2105	38
10	103	5003	49
20	115	5714	50
40	125	5797	46

# of Files	SSD (EXT3)	SSD (NTFS)	HDD (NTFS)
100만	2105	2200	56
200만	714	93	37
400만	230	86	32



- excellence in case of both small files and random read operations, but

Electronics and Telecommunications Research Institute
ETRI 한국전자통신연구원

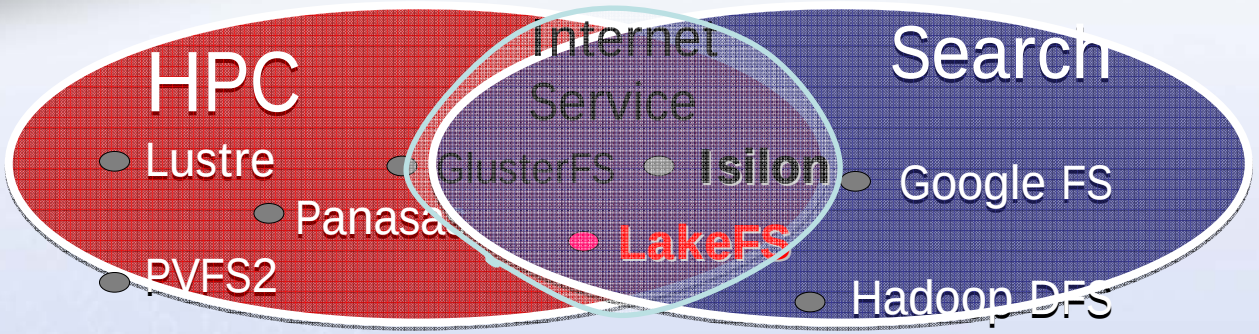
Summary

Summary

- **LakeFS**
  - a storage system for the GLORY platform
  - very large scalable distributed file system focusing on fault tolerant internet services
  - LakeFS V1.2 released!

- **Ongoing jobs (this year)**
  - clustering of file system metadata
  - intelligent storage management

LakeFS: Market Positioning



Targets	HPC	3H-NAS	Search
File System	Parallel File System	3H File System (high scalable, performance, available)	3H File Middle-ware
Workload	Parallel Read/Write	Large, Seq, Read Intensive	Large, Seq, Read/Append
Optimized for	Performance	Performance/Cost Reliability/Cost	Performance/Cost Reliability/Cost
Reliability	H/W based (SD)	S/W based (Replica)	S/W based (Replica)
Developed within	Mostly Kernel	Mostly User	User (java)
Kernel Dependency	Very High	Very Low	No
API	MPI/IO, POSIX	POSIX-FS	Native
Consistency Model	UNIX semantic, Parallel I/O semantic	NFS-semantic, Atomic Replica update	No cache Atomic replica update


감사합니다
THANK YOU
ETRI 한국전자통신연구원
Storage System Research Team