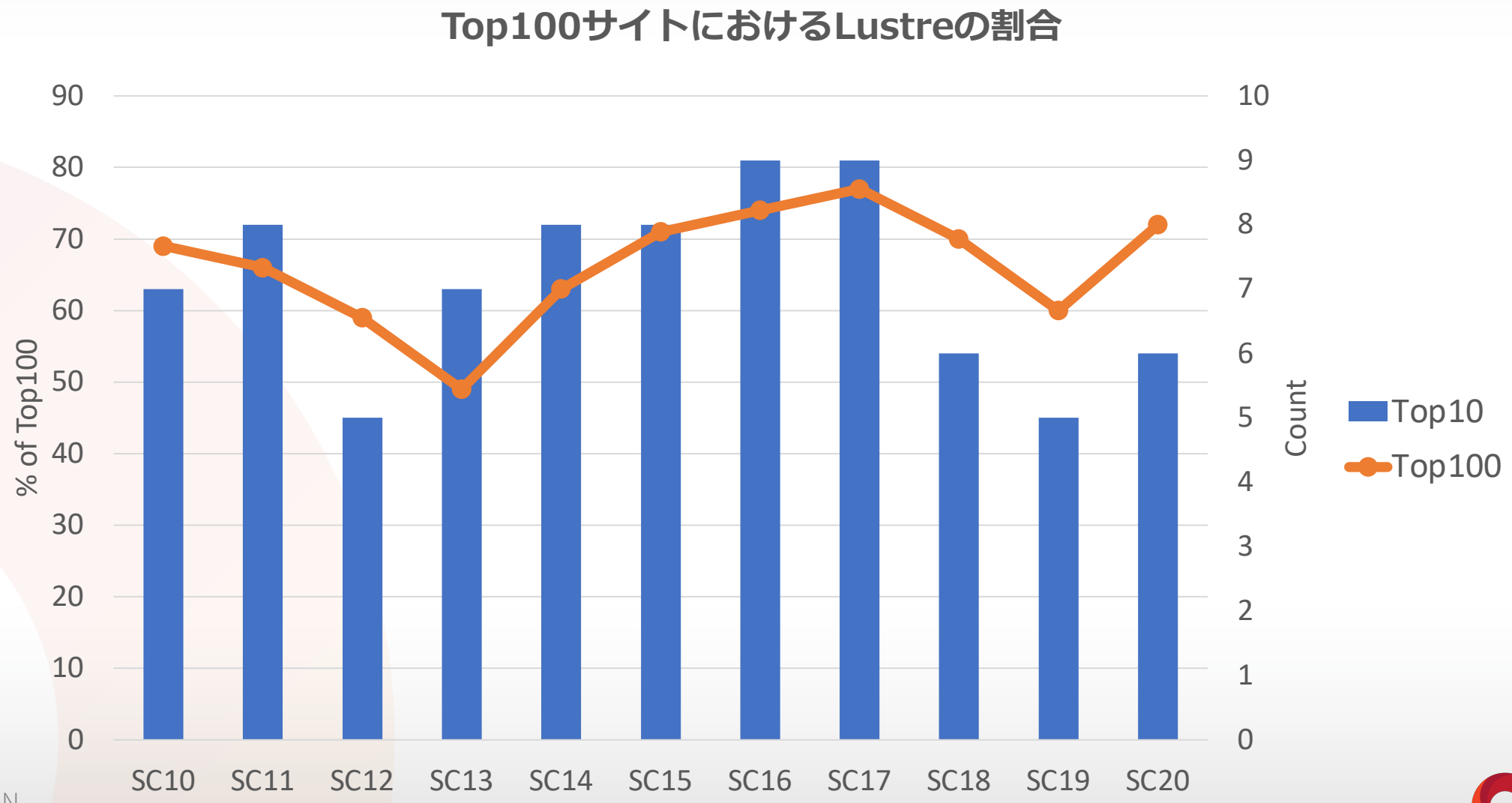
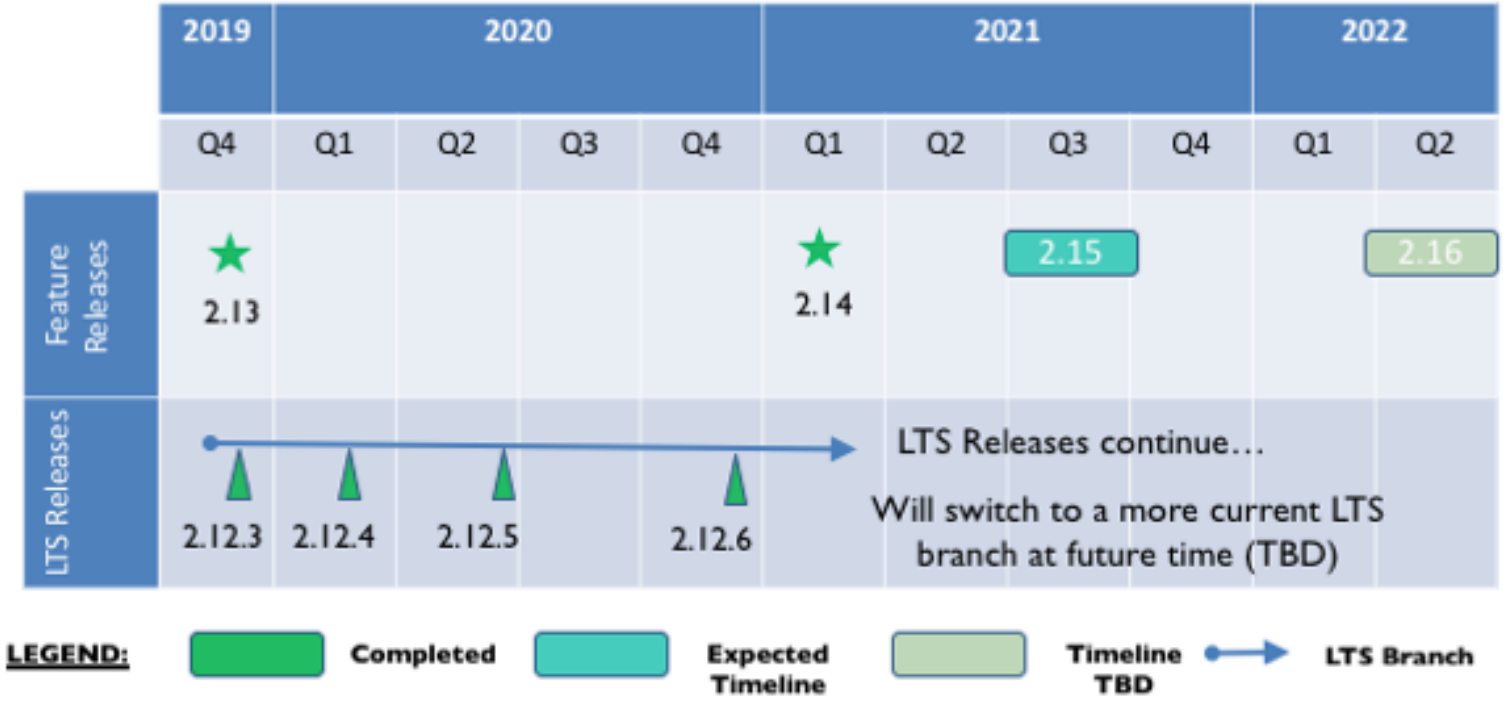
Lustreの最新機能のご紹介

DDN Storage
Shuichi Ihara
2021/03/05

Lustreは長年HPCストレージをリード



Lustre Community Roadmap



2.13
- Persistent Client Cache
- Multi-Rail Routing
- Overstriping

2.14
- Client Data Encryption
- OST Pool Quotas
- DNE Auto Restriping

2.15
- FLR Erasure Coding
- Client Directory Encryption
- LNet IPv6 Addressing

2.16
- FLR Immediate Mirror
- Metadata Writeback Cache

* Estimates are not commitments and are provided for informational purposes only
* Fuller details of features in development are available at http://wiki.lustre.org/Projects

OpenSFS

ddn

Lustre新機能の紹介

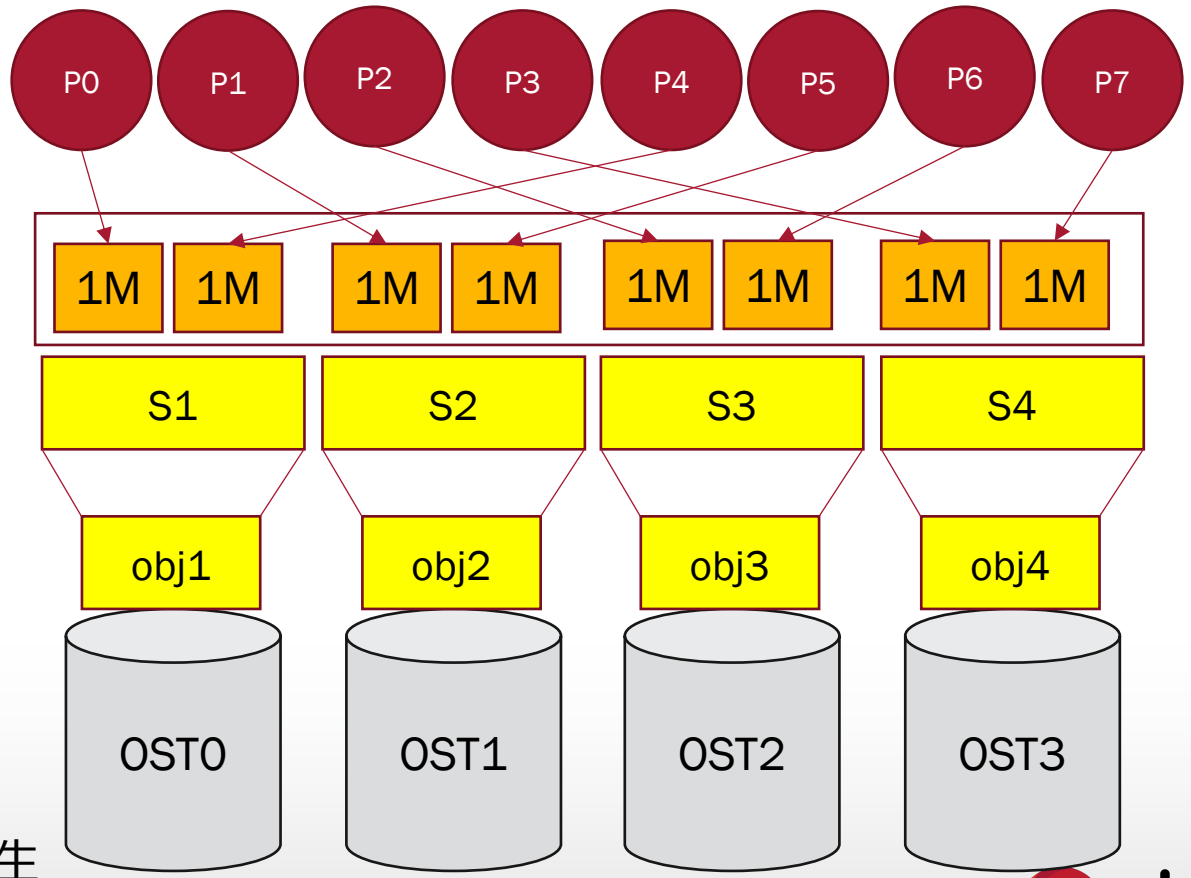
- Lustre Over Striping (Lustre-2.13)
- PCC (Persistent Client Cache) (Lustre-2.13)
- Client Data Encryption (Lustre-2.14)
- DNE Auto rebalancing (Lustre-2.14)
- OST Pool Quota (Lustre-2.14)

Lustre Over Striping

Lustre Striping

```
root@ubuntu1804-1:~# lfs setstripe -c -1 /ai200x1/shared-file
root@ubuntu1804-1:~# lfs getstripe /ai200x1/shared-file
/ai200x1/sharedfile
lmm_stripe_count:  4
lmm_stripe_size:   1048576
lmm_pattern:       raid0
lmm_layout_gen:    0
lmm_stripe_offset: 1
obdidx objid objid group
     1                 2           0x2                 0
     3                 2           0x2                 0
     0                 2           0x2                 0
     2                 2           0x2                 0
```

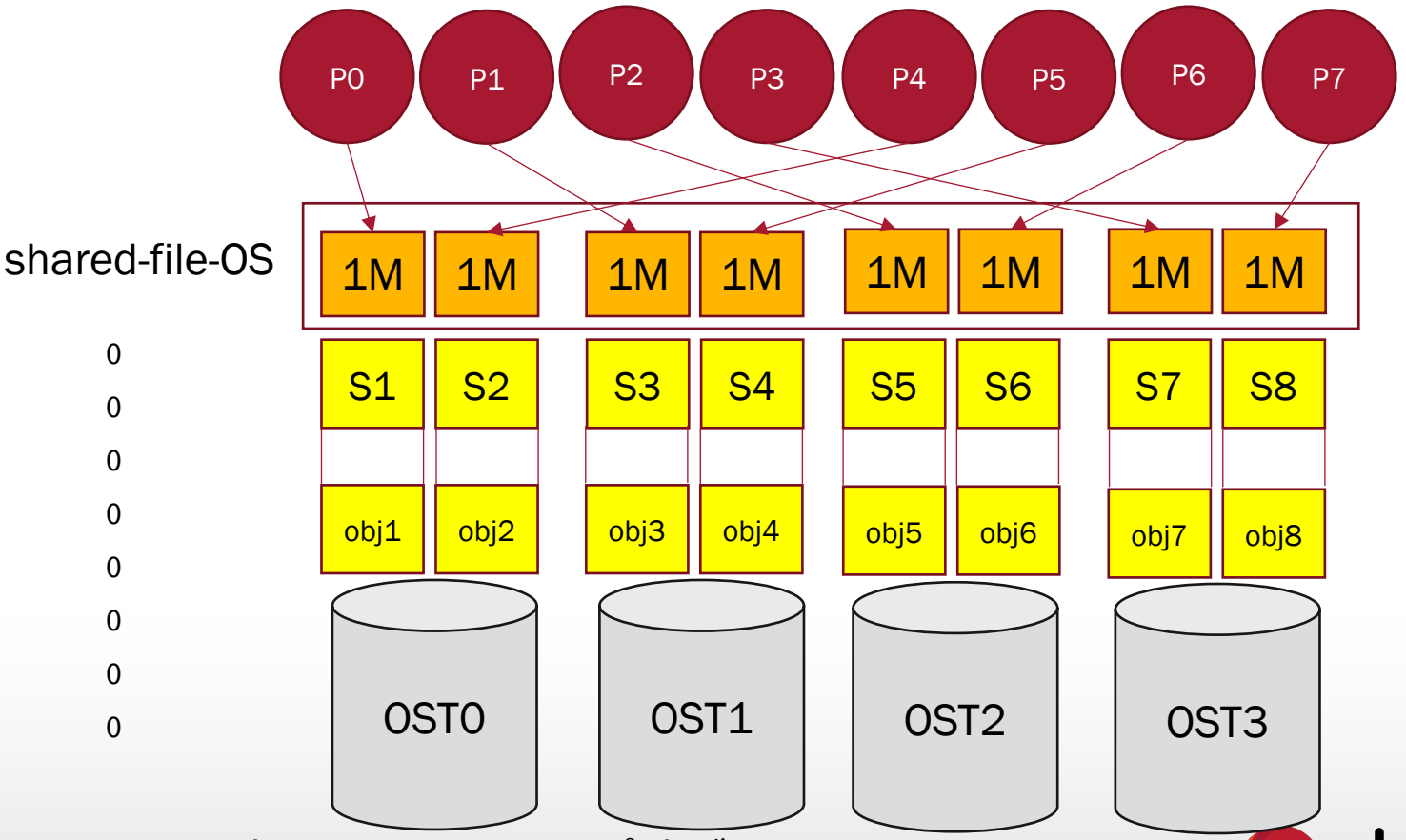
("-c -1"は全てのOST)



プロセス数 > ストライプ数の場合
OSTオブジェクトに対する競合が発生

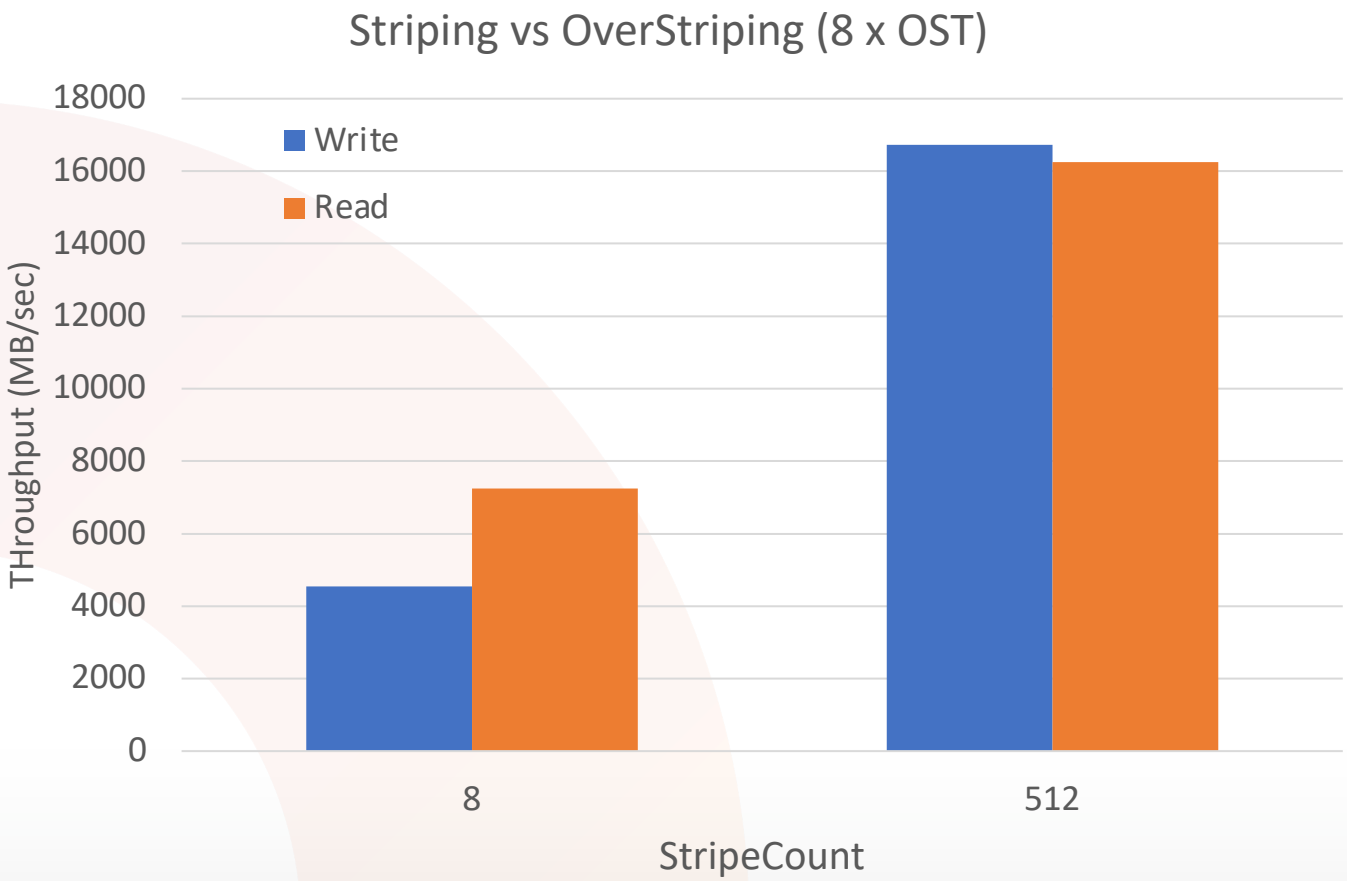
Lustre OverStriping

```
root@ubuntu1804-1:~# lfs setstripe -C 8 /ai200x1/shared-file-OS
root@ubuntu1804-1:~# lfs getstripe /ai200x1/shared-file-OS
/ai200x1/shared-file-OS
lmm_stripe_count:  8
lmm_stripe_size:   1048576
lmm_pattern:       raid0,overstriped
lmm_layout_gen:    0
lmm_stripe_offset: 0
obdidx objid objid group
     0             4          0x4                0
     2             4          0x4                0
     1             4          0x4                0
     3             4          0x4                0
     0             5          0x5                0
     2             5          0x5                0
     1             5          0x5                0
     3             5          0x5                0
```



OverStriping機能によってOST数以上のストライプ数が
作成可能になりオブジェクト内における競合を排除

Striping vs OverStriping性能比較



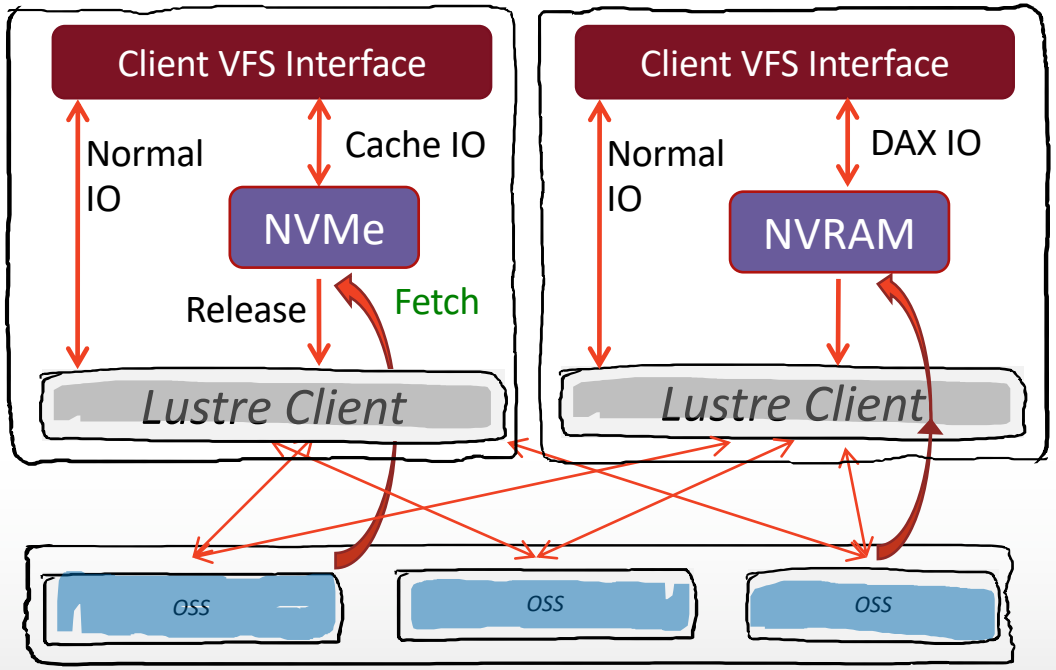
- ES7990(160 x HDD, 2 x OSS, 8 x OST)
- 32クライアント, 512プロセス
- 1MB, SingleShared File

  # ior/src/ior -w -r -C -g -i 3 -vv -s 13000 -b 1m -t 1m -a POSIX -e
- ストライプカウント8と512で比較

PCC (Persistent Client Cache)

PCC (Persistent Client Cache)

- **レイテンシの削減**, 小さくアラインしてないIOPS、ネットワークトラフィックの削減
- PCCはLustreとクライアントの**ローカルストレージ**とインテグレーション
  - **クライアントデバイス(NVMe/NVRAM)上にローカルファイルシステムを作成**
  - PCC上に作成されたファイルはグローバルネームスペースからもアクセス可能
- PCCはReadのための大容量、低レイテンシなレプリカ
- Writeは排他的なレプリカ
  - Lustre-HSMを活用
  - バックエンドのファイルシステムと同期
- DAX(Direct Access)可能なNVRAM
  デバイスとインテグレーション
  - NVRAMファイルシステム(例Nova)でさらに加速
- ReadOnlyキャッシュもサポート予定



Client Data Encryption

Lustreにおける暗号化

- ユースケース:
  - 各ユーザの特定のディレクトリに含まれるファイルに対する機密性を提供
- ゴール:
  - クライアントおよびサーバ間でデータ保護
  - 保存データの保護
- ソリューション
  - fscrypt kernel APIに準拠
    - ext4, F2FS, and UBIFSにて使われている
    - 基本原則: PageCacheに含まれるPageはクリアテキストデータを含む
  - fscrypt ユーザスペースツールを活用
- Lustreにおける実装
  - 暗号化の方法
    - LustreClientにて透過的にWrite時に暗号化、Read時に復号を実施
  - ディレクトリにおけるポリシーの適応方法
    - fscryptユーザスペースを使った新しいIOCTLのサポート
    - アトミックな暗号化コンテキストの処理

Lustre Client Encryption – ポリシーのための新しいioctls

- fscrypt ユーザスペースツール
  - Lustreにおいても改変なしで動作(thanks to fscrypt API support)
  - Encryption policies v2サポート
  - protectors (passphrase, raw key, pam)のポリシーへの紐付け

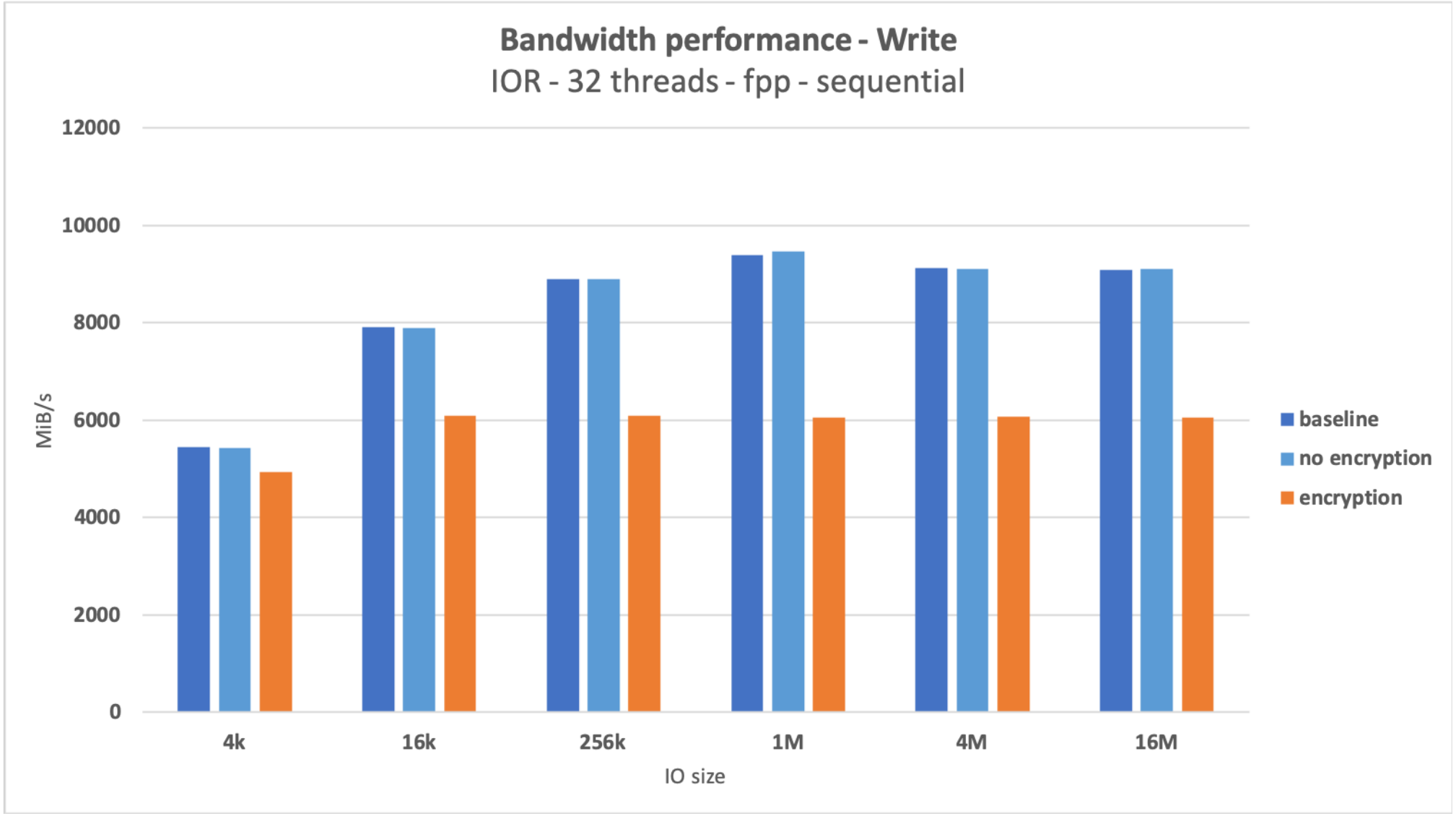
Client-A

```
root@ubuntu1804-1:~# fscrypt setup /ai200x1/
sihara@ubuntu1804-1:~$ fscrypt encrypt /ai200x1/home/sihara/encrypted
Should we create a new protector? [y/N] y
The following protector sources are available:
1 - Your login passphrase (pam_passphrase)
2 - A custom passphrase (custom_passphrase)
3 - A raw 256-bit key (raw_key)
Enter the source number for the new protector [2 -
custom_passphrase]:
Enter a name for the new protector: shield
Enter custom passphrase for protector "shield":
Confirm passphrase:
"/ai200x1/home/sihara/encrypted" is now encrypted, unlocked, and
ready for use.

sihara@ubuntu1804-1:~$ echo 12345 >
/ai200x1/home/sihara/encrypted/file
```

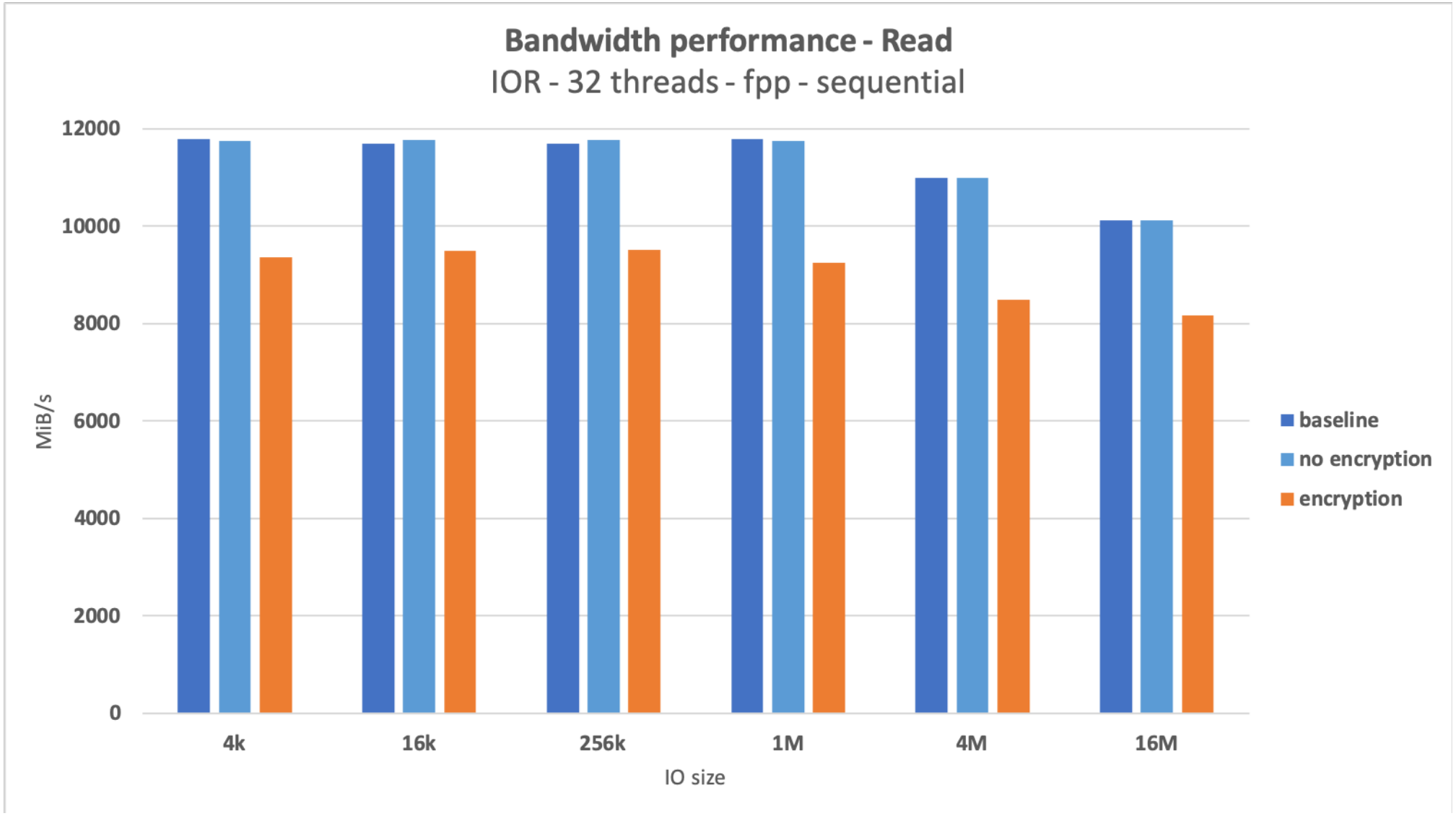
Client-B

```
king@ubuntu1804-2:~$ cat /ai200x1/home/sihara/encrypted/file
cat: /ai200x1/home/sihara/encrypted/file: Required key not available

sihara@ubuntu1804-2:~$ cat /ai200x1/home/sihara/encrypted/file
cat: /ai200x1/home/sihara/encrypted/file: Required key not available

sihara@ubuntu1804-2:~$ fscrypt status /ai200x1/home/sihara/encrypted
"/ai200x1/home/sihara/encrypted" is encrypted with fscrypt.

Policy:   4e5701b8ff6d8bfbbf7b74da199eeb88
Options:  padding:32 contents:AES_256_XTS filenames:AES_256_CTS
policy_version:2
Unlocked: No

Protected with 1 protector:
PROTECTOR          LINKED  DESCRIPTION
5dee0134a8da53b1   No      custom protector "shield”

sihara@ubuntu1804-2:~$ fscrypt unlock /ai200x1/home/sihara/encrypted
Enter custom passphrase for protector "shield":
"/ai200x1/home/sihara/encrypted" is now unlocked and ready for use.


sihara@ubuntu1804-2:~$ cat /ai200x1/home/sihara/encrypted/file
12345
```

Lustre Client Encryption – bandwidth performance



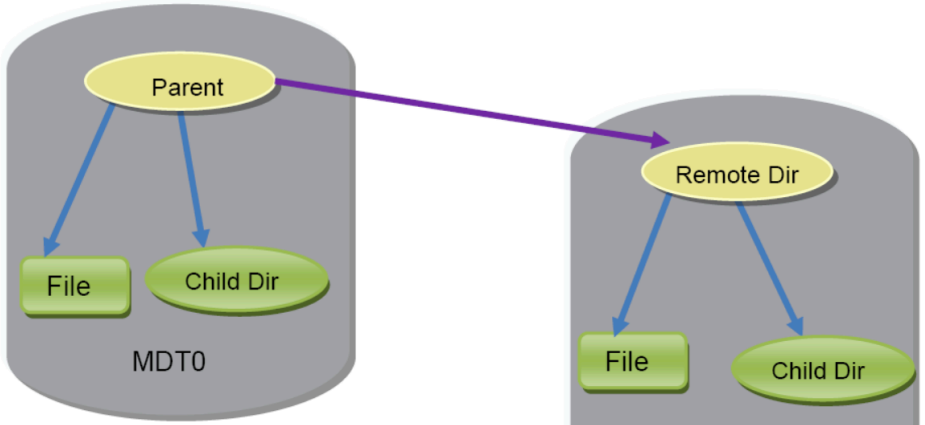
Lustre Client Encryption – bandwidth performance



DNE Auto rebalancing

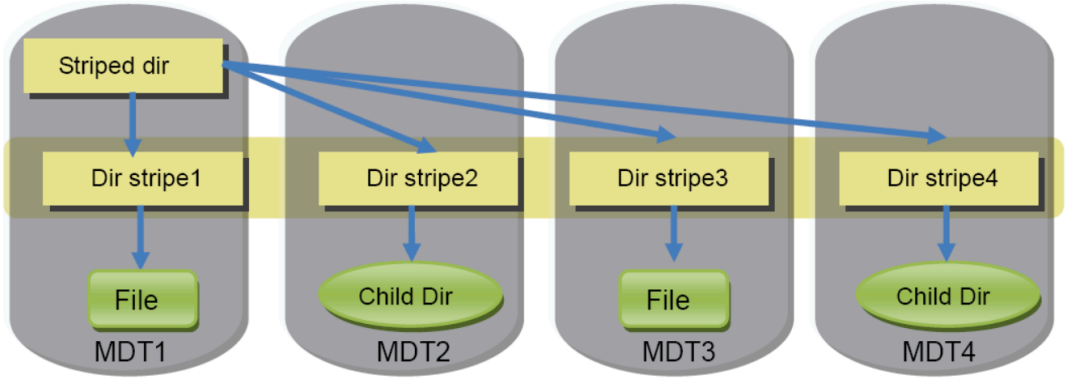
DNE(Distributed NameSpace Environment)

- Lustreは２つのタイプのDNEをサポート
  - Remote Directory(DNE1)とStriped Directory(DNE2)
  - いずれも動的に設定できDNE1とDNE2の混在も可能



Remote Directory

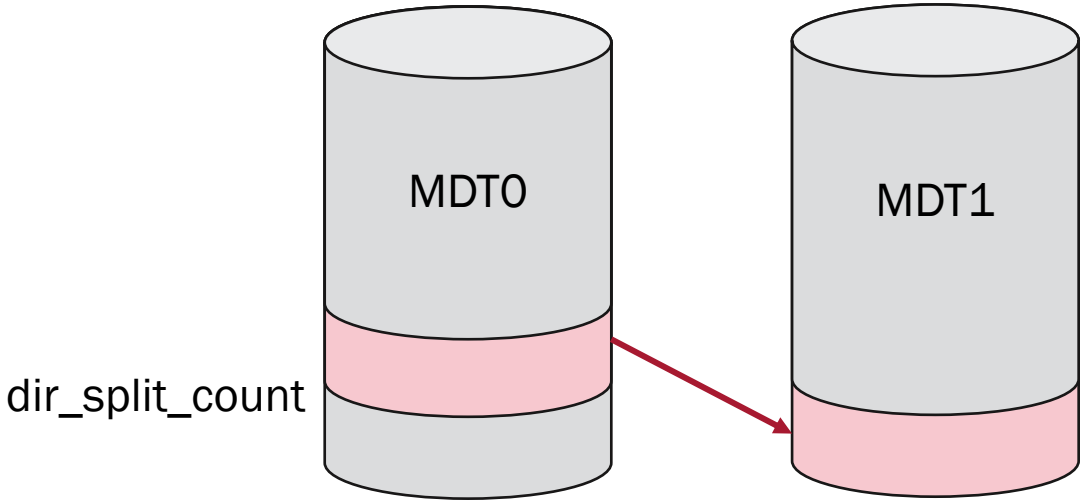
```
# lfs mkdir –i 0 /ai200x1/mdt0
# lfs mkdir –i 1 /ai200x1/mdt1
```



Striped Directory

```
# lfs mkdir –c 4 /ai200x1/striped-dir
# lfs mkdir –c 4 -D /ai200x1/ striped-dir
```

DNE Auto Rebalancing




```
[root@vexa01 ~]# lctl get_param mdt.*.enable_dir_auto_split
mdt.*.dir_split_count mdt.*.dir_split_delta
mdt.ai200x1-MDT0000.enable_dir_auto_split=1
mdt.ai200x1-MDT0000.dir_split_count=50000
mdt.ai200x1-MDT0000.dir_split_delta=2
```

enable_dir_auto_split=1(Auto rebalancing)が有効の場合、
同一ディレクトリ内でInode数がdir_split_countを超えた時点から
dir_split_deltaに基づき複数のMDTに自動的に分散される

閾値を設定することでディレクトリのサイズが大きくなる前に
複数のMDTに自動的に分散すること性能劣化を防ぐ

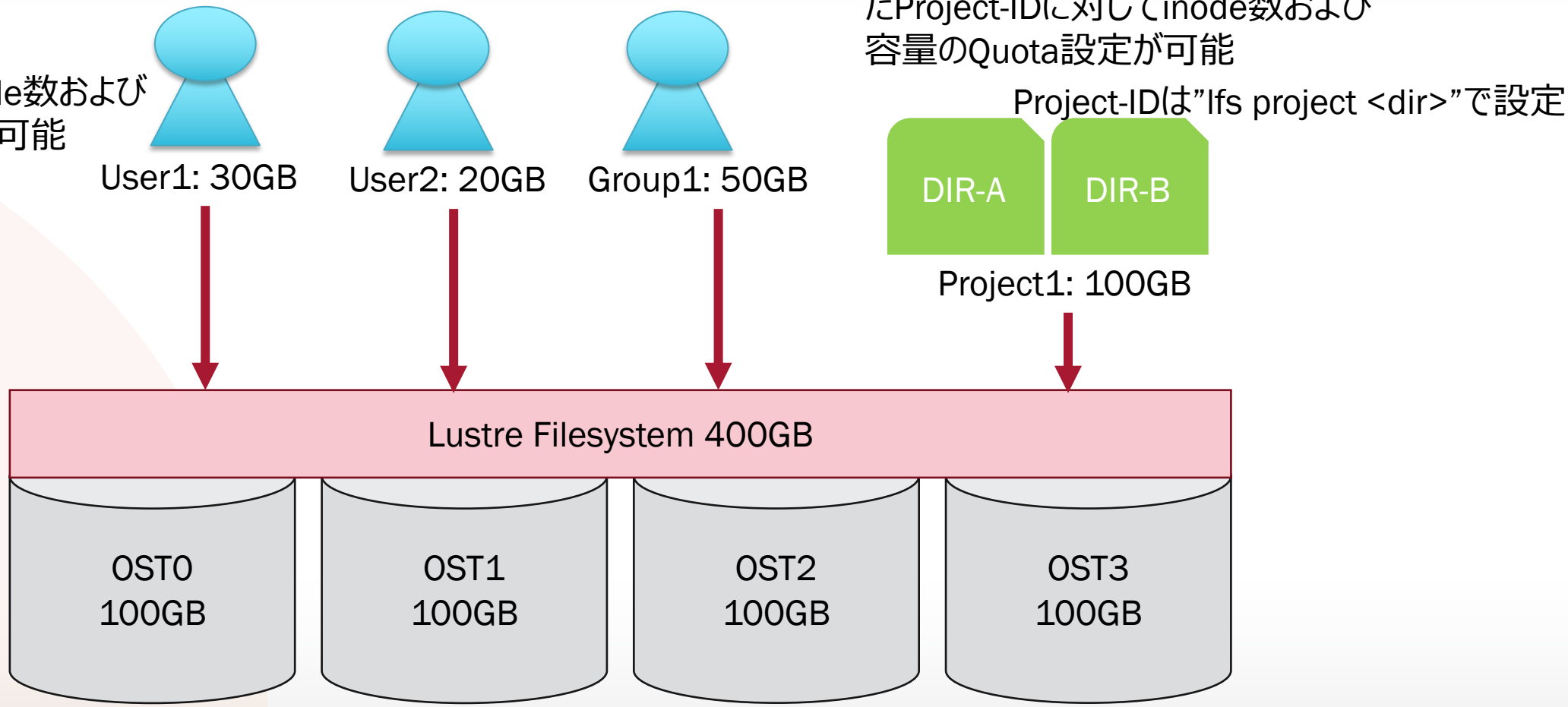
OST Pool Quota

Lustre Quota

UID, GIDに対してinode数および
容量の Quota設定が可能

特定のディレクトリファイルに付与されたProject-IDに対してinode数および容量のQuota設定が可能

Project-IDは”lfs project <dir>”で設定



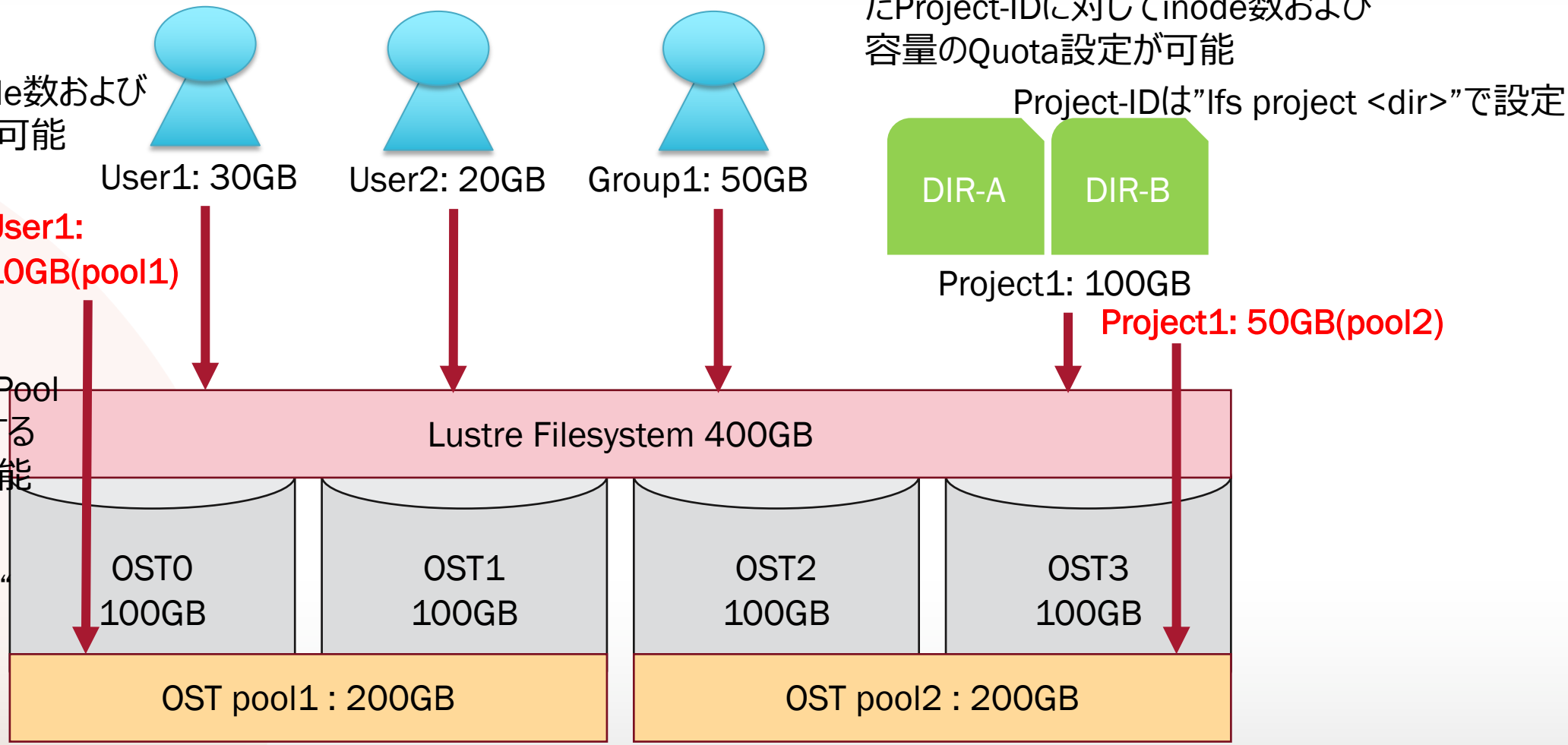
いずれもファイルシステム全体に対するQuota設定。
特定OST群を対象とした領域の設定ができない。
例) HDDとNVMeのOSTが混在した際、それぞれの領域に異なるQuota設定

OST Pool Quota

UID, GIDに対してinode数および
容量の Quota設定が可能

特定のディレクトリファイルに付与されたProject-IDに対してinode数および容量のQuota設定が可能

Project-IDは”lfs project <dir>”で設定



事前に作成されたOST Poolに対してUID, GIDに対する容量のQuota設定が可能

OST Poolは”lctl pool_create/pool_add“で動的に作成

1 つのファイルシステムに混在する異なるデバイス(HDD, NVMe)毎にそれぞれOST Poolを作成して、異なるQuota設定

ddn