2022, © NTT All Rights Reserved

[JLUG 2022]

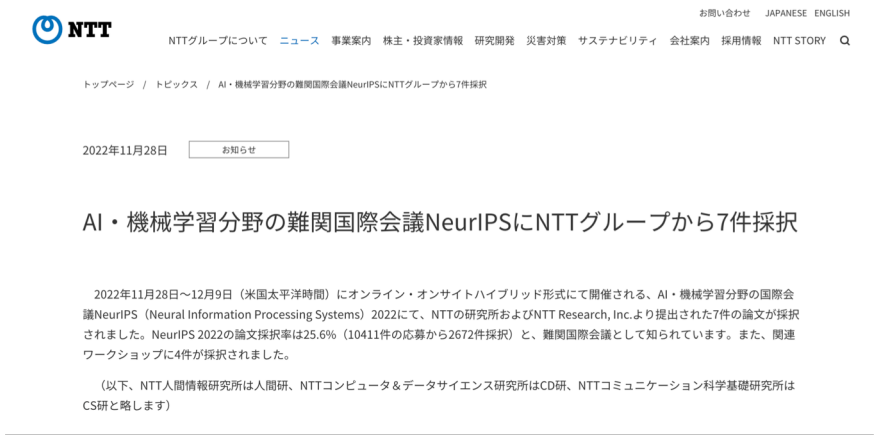
Lustreを用いたストレージ高速化に関するNTTの取り組み

NTT コンピュータ＆データサイエンス研究所

露崎　浩太

NTTの研究開発

- 4総合研究所、14 研究所 [1]
  - IOWN構想[2]、コミュニケーション、ネットワーク、デバイスなどの基礎技術
- AI/機械学習の分野で高い研究開発成果
  - これらの計算基盤などについても研究開発



お問い合わせ　JAPANESE　ENGLISH

NTT　NTTグループについて　ニュース　事業案内　株主・投資家情報　研究開発　災害対策　サステナビリティ　会社案内　採用情報　NTT STORY

トップページ / トピックス / AI・機械学習分野の難関国際会議NeurIPSにNTTグループから7件採択

2022年11月28日　お知らせ

AI・機械学習分野の難関国際会議NeurIPSにNTTグループから7件採択

2022年11月28日～12月9日（米国太平洋時間）にオンライン・オンサイトハイブリッド形式にて開催される、AI・機械学習分野の国際会議NeurIPS（Neural Information Processing Systems）2022にて、NTTの研究所およびNTT Research, Inc.より提出された7件の論文が採択されました。NeurIPS 2022の論文採択率は25.6%（10411件の応募から2672件採択）と、難関国際会議として知られています。また、関連ワークショップに4件が採択されました。

（以下、NTT人間情報研究所は人間研、NTTコンピュータ＆データサイエンス研究所はCD研、NTTコミュニケーション科学基礎研究所はCS研と略します）

https://group.ntt/jp/topics/2022/11/28/neurips_2022_ntt.html

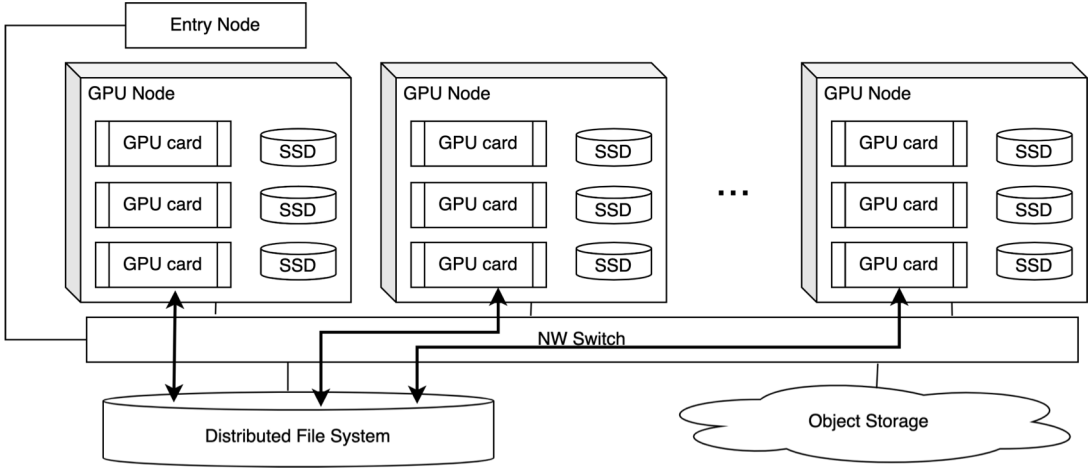
1: https://www.rd.ntt/organization/laboratory/
2: https://www.rd.ntt/iown/

自己紹介

- 氏名：露崎 浩太
- 所属：コンピュータ&データサイエンス研究所
  革新的コンピューティングアーキテクチャ研究プロジェクト
- バックグラウンド
  - OpenStack Swift (Object Storage)の研究開発 (コアデベロッパ)
  - 社内AI/ML基盤の開発/運用
    - GTC21: https://www.nvidia.com/en-us/on-demand/session/gtcspring21-s31410/
  - 次世代データ処理基盤の研究開発（現職）
    - Lustreに関する高速化/利便性向上に関する技術開発、検討
    - JLUG21: https://www.jlug.info/_files/ugd/8d1181_516f4ba25dda4979a2ade162752a24c2.pdf

AI/ML基盤: データ転送パスの課題

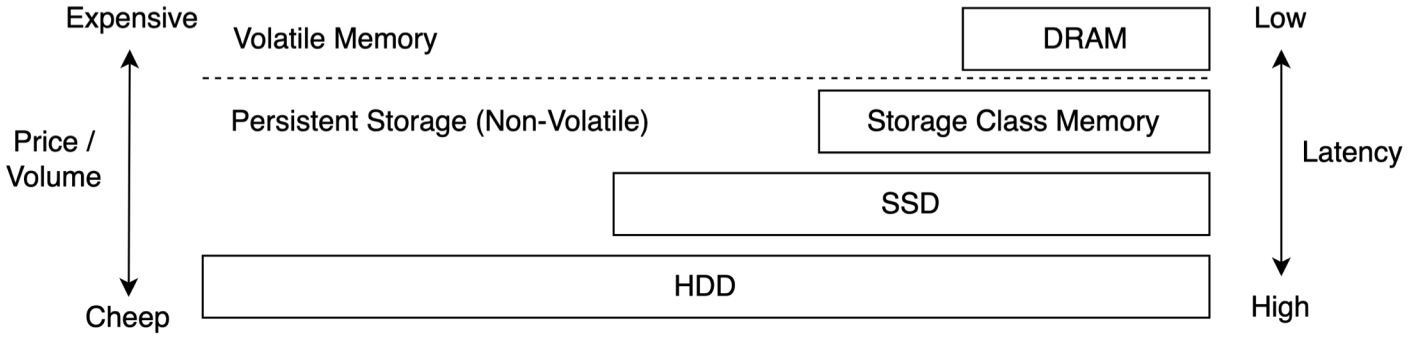
- ユーザ向けのデータ保存領域としてNW接続された分散ストレージを提供
- ストレージ、計算ノード間のデータ転送
  - 広帯域な転送レート、計算ノードに対する輻輳の考慮
  - 計算ノードで生成されるデータのストレージへの書き込み



GPUクラスタの構成例

階層化ストレージ

- 速度/容量/コスト要件に対する最適化のために異なる性質のデバイスを使い分ける技術
  - よく使うデータ: 高速/小容量/高価のデバイス
  - あまり使わないデータ: 低速/大容量/安価のデバイス
- デバイス間だけでなく別ストレージシステムと連携する階層化もある
  - 特定のノードグループ用のストレージレイヤーや計算ストレージ内のデバイスへの階層化 (富嶽など *1)
  - S3など他のストレージシステムとの階層化 (AWS FSx for Lustreなど *2)



*1: https://www.jlug.info/_files/ugd/8d1181_f4c8a5e4774940d09674d175de5def4b.pdf
*2: https://docs.aws.amazon.com/ja_jp/fsx/latest/LustreGuide/exporting-files-hsm.html

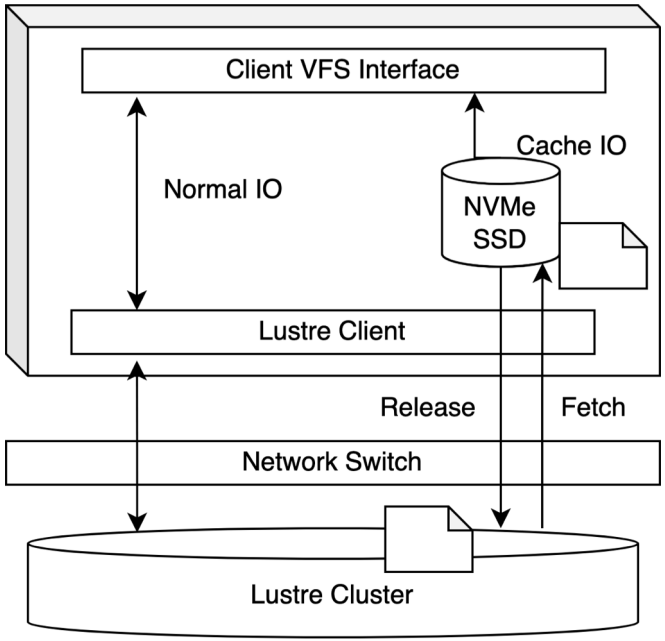
Lustre Filesystem

- Open Source Parallel File System
  - GPLv2 License, POSIX互換
- TOP500のクラスタで多く利用されている
  - IO500でもエントリ
- スケーラブルな性能/容量
  - PBクラスのプロダクション利用あり
- Information
  - https://www.lustre.org/
  - https://wiki.lustre.org/Main_Page
  - https://www.opensfs.org/events/lug-2022/

Lustre Persistent Client Cache

コミュニティ

- Lustre Client Software 2.14からリリースされた機能
- Lustre Client ToolでLocalのCache Deviceを追加
  - Lustre上でのファイルに対する透過的なアクセス
  - Linux上で扱えるデバイスであれば利用可能
- Read-Write/Read-onlyの2つのモード
- 自動Cacheもしくは明示的なCache操作が必要



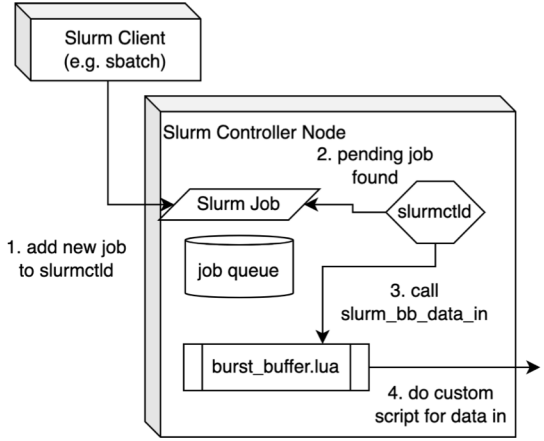
Slurm Workload Manager

- ジョブスケジューラ
- https://slurm.schedmd.com/
- GPL v2 License
- 5 systems of top 10 (of TOP 500)
  - 2020, June Statistics
  - https://sc20.supercomputing.org/proceedings/bof/bof_pages/bof102.html

Slurm Burst Buffer Lua Plugin

コミュニティ

- Generic Burst Buffer Interface for Slurm Workload Manager
  - Since Slurm Workload Manager 21.08 (https://slurm.schedmd.com/burst_buffer.html)
- AdminがJob State毎に処理を挿入することができる仕組み
  - Prolog/Epilogより広いJob Stateをサポート



example: slurm_bb_data_in

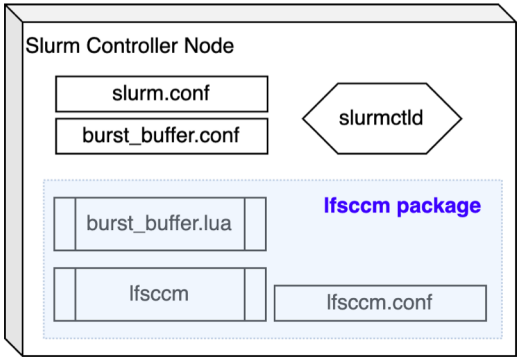
Job State	burst_buffer.lua function
Job submission	slurm_bb_job_process
Job pending	slurm_bb_data_in
Job allocated resources not running yet	slurm_bb_pre_run
Job finished, completing state	slurm_bb_data_out
Job completed	slurm_bb_teardown

https://youtu.be/q5j5meDDPpA
https://slurm.schedmd.com/SLUG21/burst_buffer_lua.pdf

lfsccm: Burst Buffer Lua for Lustre

NTT

- lfsccm: Lustre File System Client Cache Manager
  - https://github.com/DDNStorage/lfsccm
- OSSエコシステム
  - Slurm Controller Nodeへ lfsccm パッケージをインストールするだけで使える
  - キャッシュの仕組みはLustre の Persistent Client Cache (*1) をそのまま利用
  - Slurmと似たsyntaxの設定ファイル



Install:
  pip install lfsccm

lfsccm.conf:
NodeName=<compute_node> rwid=<cache id> roid=<cache id>

*1: https://doc.lustre.org/lustre_manual.xhtml#pcc

lfsccm: How to Use

NTT

- SlurmのBatchスクリプトに “#PCC” ディレクティブで設定

main / lfsccm / lustre-vm / sample_jobs / fetch.sh Go to file ···

bloodeagle40234 Add lfsccm project files Latest commit bcb75cd on Sep 9 History

1 contributor

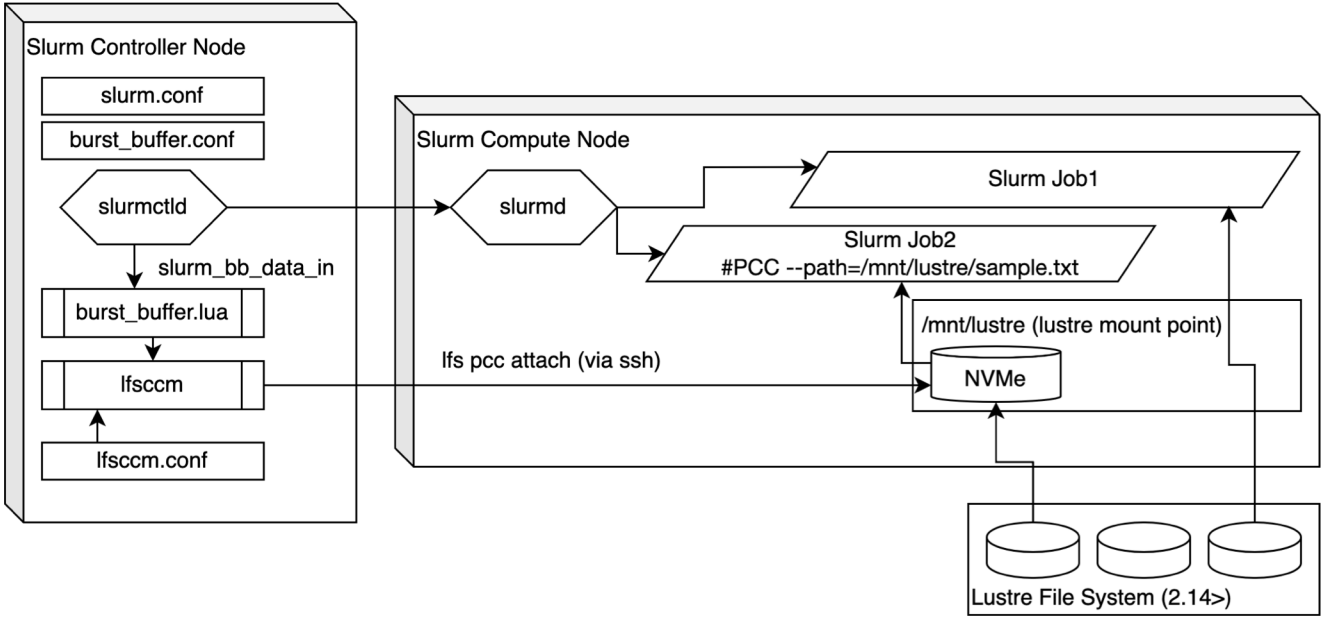
5 lines (4 sloc) | 110 Bytes Raw Blame

```
1  #!/bin/bash
2  #PCC --path=/mnt/lustre/sample --mode=rw
3
4  cat /mnt/lustre/sample
5  lfs pcc state /mnt/lustre/sample
```

https://github.com/DDNStorage/lfsccm/blob/main/lustre-vm/sample_jobs/fetch.sh

lfsccm: How it works

- Slurm Burst Bufferの slurm_bb_data_in のタイミングで LustreのPCCを使ったprefetchを実施

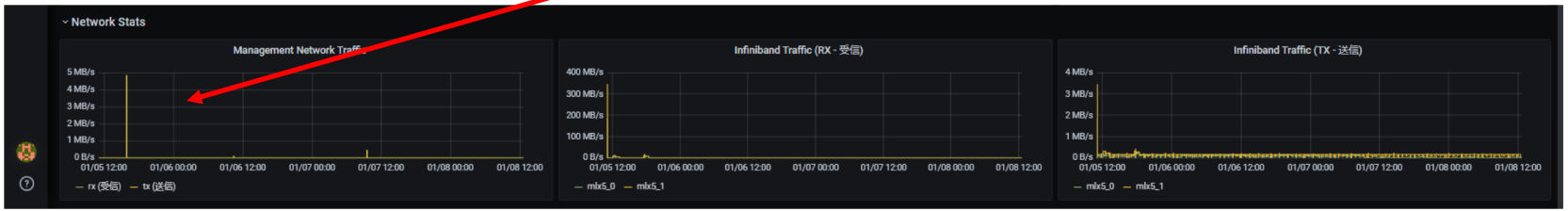


Performance Summary: LustreのIO調査

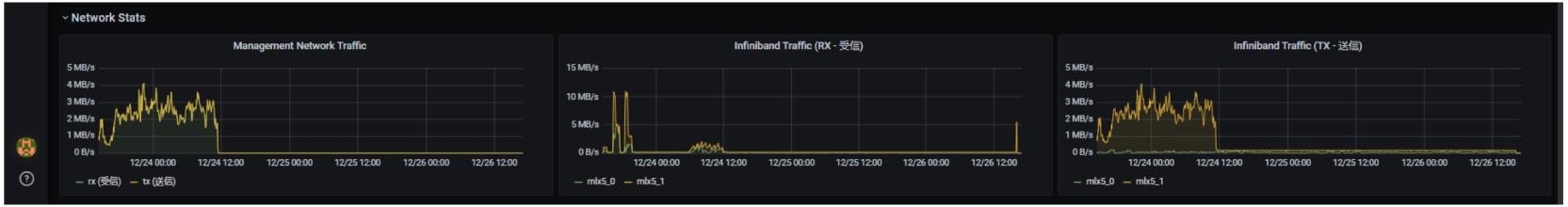
ベンチマーク: Large Scale GANの学習

Lustre Traffic (lfsccm ON)

“data in”のタイミングで1回だけキャッシュデバイスへのシーケンシャルリードが発生



Lustre Traffic (lfsccm OFF)



lfsccm: 制限事項

NTT

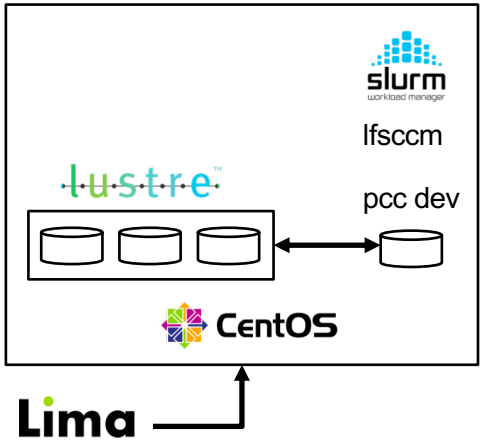
- 利用できる読み書きモードがLustreのバージョンに依存
  - Lustre 2.14の時点ではRWモードのみをサポート
  - ROモードはLustre 2.16のリリース候補
  - lfsccm は LustreのDevelopment Branchを利用してRO/RWの両方をテスト済み
- Invalid Configに対する保守的な挙動
  - Slurm BB Pluginの制約でWarning出力して実行を許容するような作りができない
  - 基本的に無効な設定を検出した場合はジョブ自体が失敗する

lfsccm: Demo and Testing

NTT

- Demo/Testing
  - https://github.com/DDNStorage/lfsccm/blob/main/demo.sh
  - Slurm/Lustre/lfsccmをセットアップするAll-In-OneのVM環境
    - VMの利用にLima (https://github.com/lima-vm/lima) が必要
  - `demo.sh` のコマンドで環境構築可能
    - lustre-vm/lustre.yaml # lima project file for VM
    - lustre-vm/lustre/* # Lustre setup scripts
    - lustre-vm/slurm/* # Slurm and BB lua for lustre setup scripts

demo環境構成



lfsccm: Demo

Slurm User Group 2022:
https://www.youtube.com/watch?v=jyUQ33GTnnE

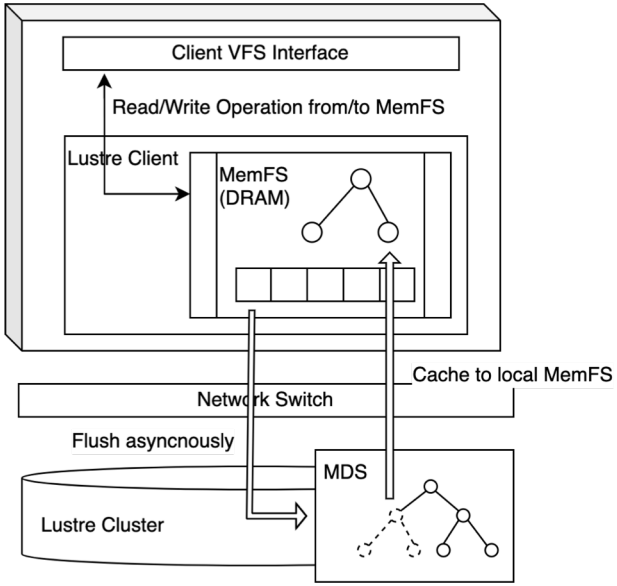
```
yukimura:lfsccm tsuyuzaki$ limactl shell lustre sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug*       up   infinite      1   idle lima-lustre
yukimura:lfsccm tsuyuzaki$ limactl shell lustre scontrol show node=lima-lustre
NodeName=lima-lustre Arch=x86_64 CoresPerSocket=4
   CPUAlloc=0 CPUTot=4 CPULoad=0.00
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=(null)
   NodeAddr=lima-lustre NodeHostName=lima-lustre Version=21.08.8-2
   OS=Linux 4.18.0-240.1.1.el8_lustre.x86_64 #1 SMP Fri Feb 19 20:41:01 UTC 2021
   RealMemory=1 AllocMem=0 FreeMem=1462 Sockets=1 Boards=1
   State=IDLE ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=debug
   BootTime=2022-09-12T01:38:36 SlurmdStartTime=2022-09-12T06:13:44
   LastBusyTime=2022-09-12T06:13:54
   CfgTRES=cpu=4,mem=1M,billing=4
   AllocTRES=
   CapWatts=n/a
   CurrentWatts=0 AveWatts=0
   ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s

yukimura:lfsccm tsuyuzaki$ limactl shell lustre lctl lustre_build_version
Lustre version: 2.14.0
yukimura:lfsccm tsuyuzaki$ cat lustre-vm/sample_jobs/fetch.sh
#!/bin/bash
#PCC --path=/mnt/lustre/sample --mode=rw

cat /mnt/lustre/sample
lfs pcc state /mnt/lustre/sample
yukimura:lfsccm tsuyuzaki$ limactl shell lustre bash -c "sbatch -o ~/result lustre-vm/sample_jobs/fetch.sh"
Submitted batch job 22
yukimura:lfsccm tsuyuzaki$ limactl shell lustre squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
yukimura:lfsccm tsuyuzaki$ limactl shell lustre bash -c "cat ~/result"
hello burst buffer
file: /mnt/lustre/sample, type: readwrite, PCC file: /0001/0000/0401/0000/0002/0000/0x200000401:0x1:0x0, user number: 0, flags: 0
yukimura:lfsccm tsuyuzaki$
```

NTT

Lustre Writeback Cache

コミュニティ

- LustreのNamespace Treeの一部をClient側にキャッシュ
- ファイルの大量書き込みなどMDSへのアクセスの多いワークロードの改善
- Lustre 2.17 (2024 Q1) ターゲット
  - LAD 22 Community Updates(*1)より
- Integrationやクラスタ設計方法について検討中



paper: MetaWBC: POSIX-Compliant Metadata Write-Back Caching for Distributed File Systems
Yingjin Qian, et al.
develop: https://jira.whamcloud.com/browse/LU-10938
*1: https://www.eofs.eu/_media/events/lad22/1.1-community_release_update.pdf

まとめ

- Lustreを用いたAI/ML基盤向けのストレージ研究開発
- AI/ML基盤向けのデータパス最適化
  - 読み込み： Slurm/Lustre/lfsccmを用いた計算ノードへのデータプリフェッチ
  - 書き込み: Lustre Writeback Cacheを用いたデータ書き出しの高速化
- lfsccmの紹介

ご清聴ありがとうございました

E-mail:
Kota Tsuyuzaki (kota.tsuyuzaki.pc@hco.ntt.co.jp)
Repository:
https://github.com/DDNStorage/lfsccm