安くて大容量のNFSが欲しいと言われた
とあるインフラエンジニアの憂鬱

Kei Kusunoki
Technical Development

NTT Communications
Transform. Transcend.

自己紹介

- NTTコミュニケーションズ 技術開発部
  - NTTコムのR&Dセクション
    - ストレージ担当
- 業務
  - ~2016: Enterprise Cloud 2.0 のストレージ開発
  - Now: ストレージ検証, 新規ストレージサービスのPoC

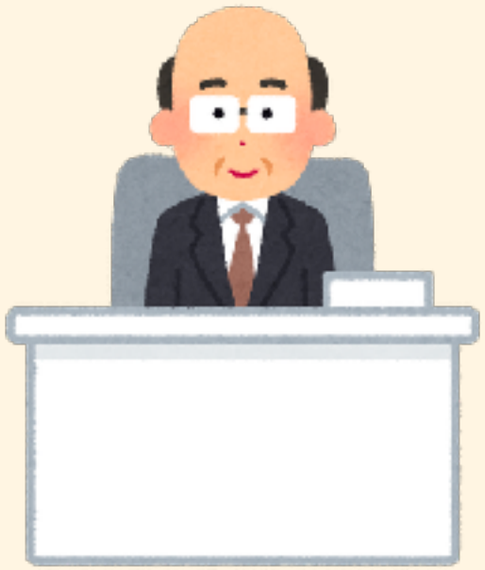
NTT Communications
Transform. Transcend.

皆さん、NFS使っていますか？

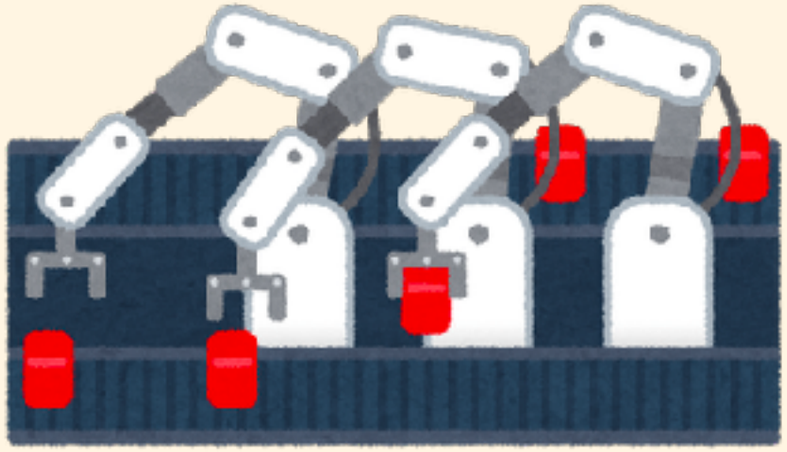
- 最近のアプリはObject Storage使える
- だがしかし
- 昔のアプリは・・・？
- 人間がマニュアルで使うのは・・・？

NTT Communications
Transform. Transcend.

高まる”ファイル”ストレージ需要



人工知能欲しい



IoT/ビッグデータ

既存アプリ改修したくない

Transform. Transcend.

従来のNFS

- 長年の実績ある専用のストレージアプライアンス
- きめ細やかな設定が可能
- だいたいスケールアップ型

スケールアップ型の辛み

- 一回の設備投資が大きいし、納期が長い
  - 常に十分な空き容量を確保しての運用
  - 歩留まりが悪い
- ビッグデータ案件だ！ 100TBよろしく頼むよ
  - Oh...

NTT Communications
Transform. Transcend.

増え続ける安い
大容量ストレージの実現のためには

- スケールアウト型の方が幸せ

- スケールアウト型ファイルストレージ
  - Isilon, FlashBlade etc.

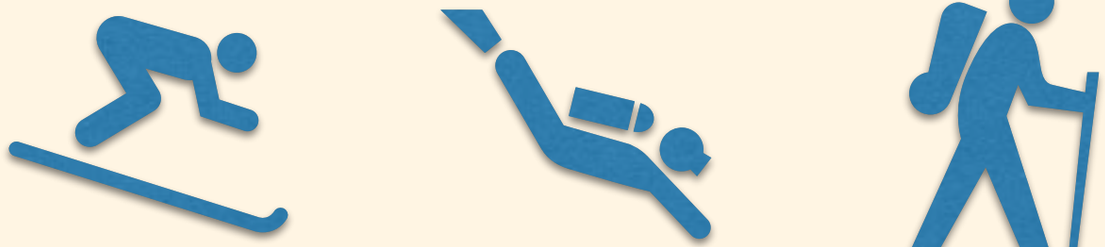
スケールアウトな
安いストレージと言えば

- オブジェクトストレージ！

- 他にも大容量が必要なバックアップ系アプリはS3がデファクト

- 大きなオブジェクトストレージを作り、それをNFSとしても使えれば一石二鳥では🤔

多彩なお客様



NFS Access

Object IFをNFSに変換する
何か

REST API

Object Node1

Object Node2

Object Node3

・・・

Object Node X

オブジェクトストレージ

オブジェクトをバックエンドにNFSする愉快な仲間達

- オブジェクトストレージ製品付属ソフト
  - NetApp StorageGRID, Scality, EMC ECS etc.
- ストレージGWソフト
  - Panzura, Fobas, Ctera, Nasuni
- Open Source
  - Ceph, Swift-on-file

NTT Communications
Transform. Transcend.

ん、待てよ・・・？
NFSとObjectって全然違う・・・

- NFSは結構複雑
  - 複雑なMETA操作: ディレクトリ構造/ファイル名処理/ACL etc
  - 部分アクセス: ファイルの一部だけの修正書き込みなど
- Objectはシンプル
  - 基本はシーケンシャルアクセスなPUT/GETだけ

NTT Communications
Transform. Transcend.

シンプルなObjectの上で複雑なNFSを実現するには 1

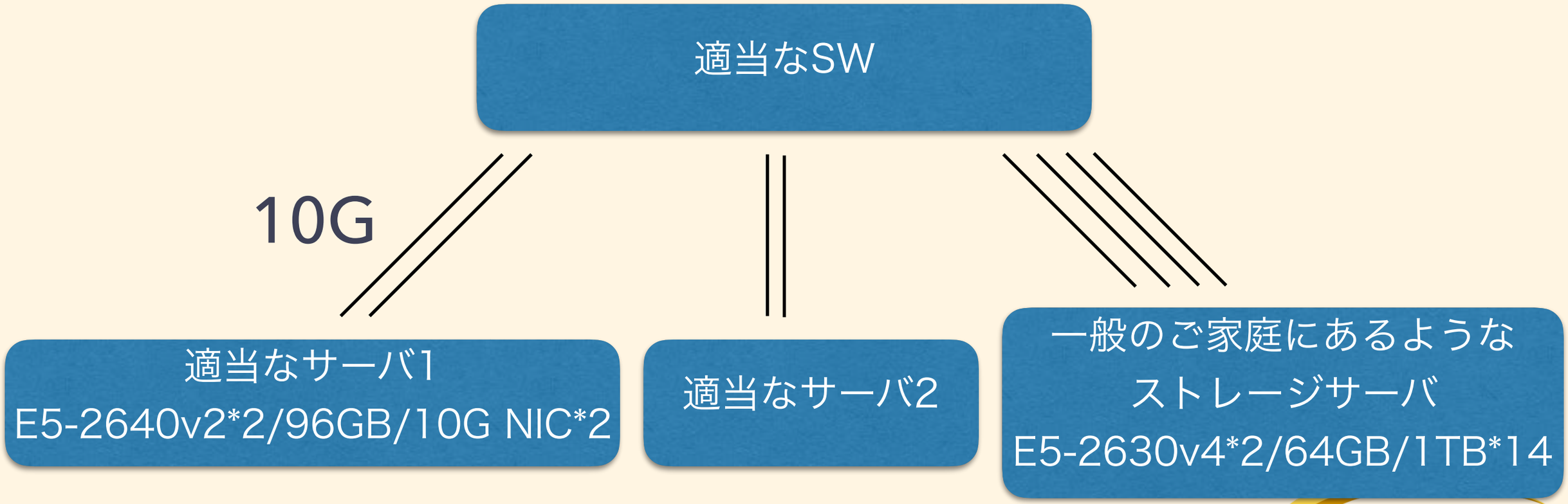
- NFSのMETA情報どうするか
    - Objectに置く
        - 一貫性あるし、GWが死んでもOKで安い
- キャッシュする
    - GWが死んだら困るので冗長化、高い
    - 一貫性やスケーラビリティは？

シンプルなObjectの上で複雑なNFSを実現するには 2

- 部分アクセスどうするか
    - 全部Objectに上書き/読み込みしてACK
    - キャッシュ

どのくらい遅くなるのか？

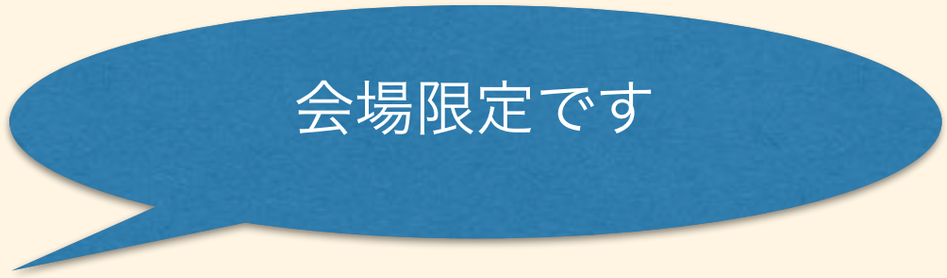
- やってみました
- 構成



*6

どのくらい遅くなるのか？

- やってみました



NTT Communications
Transform. Transcend.

書き込みがとっても遅い

- ランダムアクセスは分かるけど、シーケンシャルなのに遅い
- ベンチマークソフト変えても同じ傾向

FIO

会場限定です

NTT Communications
Transform. Transcend.

NFSの書き込みプロトコル

3.3.7 Procedure 7: WRITE - Write to file

SYNOPSIS

```
WRITE3res NFSPROC3_WRITE(WRITE3args) = 7;

enum stable_how {
     UNSTABLE  = 0,
     DATA_SYNC = 1,
     FILE_SYNC = 2
};

struct WRITE3args {
     nfs_fh3     file;
     offset3     offset;
     count3      count;
     stable_how  stable;
     opaque      data<>;
};

struct WRITE3resok {
     wcc_data    file_wcc;
     count3      count;
     stable_how  committed;
     writeverf3  verf;
};

struct WRITE3resfail {
     wcc_data    file_wcc;
};

union WRITE3res switch (nfsstat3 status) {
case NFS3_OK:
     WRITE3resok    resok;
default:
     WRITE3resfail  resfail;
};
```

https://tools.ietf.org/html/rfc1813#section-3.3.7

NTT Communications
Transform. Transcend.

NFS書き込みの実装

- クライアントからのStable Option
  - FILE_SYNC: METAも全部書いてからACK
  - DATA_SYNC: データだけ書いてからACK
  - UNSTABLE: コミットが来るまで遅らせても良い

全体図

Unstable...Commit

ACK

ストレージ GW

コミットが来る度に書き込み

Object Storage

Object全部書いてACK

結論

- シーケンシャルリードはObjectそのものと比べても遜色ない
- Writeは要注意
  - シーケンシャルでも、書き込みアプリの実装によっては非常に遅い
  - ストレージGWのパラメータチューニング、設計はとても大事
    - キャッシュ(同期/非同期)、冗長化

NTT Communications
Transform. Transcend.

用法用量を守って正しくお使いください

- Good Use-case
  - マイグレーション/バックアップ/特定用途のアプリ
- Interesting Use-case
  - “普通”のストレージだと思って使うアプリ
  - 人間が使うファイル共有

NTT Communications
Transform. Transcend.

出典:

いらすとや様

http://www.irasutoya.com/

MtU様 ちびゆかりん立ち絵素材

http://seiga.nicovideo.jp/seiga/im6624528