NICTにおける
大容量ファイルの遠隔バックアップ
実施事例の共有

2022年10月14日(金)

国立研究開発法人情報通信研究機構
脳情報通信融合研究センター
横山 輝明 /横濱 則也

National Institute of Information and Communications Technology

CiNet 脳情報通信融合研究センター Center for Information and Neural Networks

国立研究開発法人情報通信研究機構
脳情報通信融合研究センター

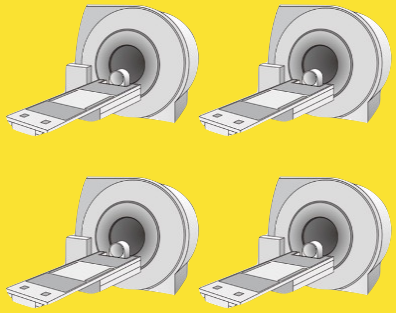
- NICT拠点のひとつ、大阪大学吹田キャンパス内
- 脳情報科学の研究を推進

脳情報通信融合研究センター(CiNet: Center for Information and Neural Networks)は、大阪・吹田市を拠点とし、異分野融合により脳情報科学の研究を進めています。CiNet研究棟は、2013年3月に開所し、最新の設備で脳の機能についての基礎研究を進めると同時に、情報通信技術、ブレイン・マシン・インターフェース、脳機能計測、ロボット工学などの相互に関連する分野での応用研究も実施しています。（https://cinet.jp/japanese/）

NICT

CiNet
脳情報通信融合研究センター
Center for Information and Neural Networks

CiNet計算機・ストレージの利用目的



脳計測装置
（7T-MRI, 3T-MRI×3, MEG）

脳計測データ
（2000件/年）

- 脳データの一元管理
- 脳データの共有
- 脳データ解析・処理
- 脳機能モデリング
- 脳型AI開発
- etc

CiNet計算機・ストレージ環境
（ペタバイト級ストレージ,
大規模計算サーバー）

脳情報
デコーディング

脳型人工知能

BMI・医療応用

AI×脳科学
融合技術

CiNet計算機・ストレージの利用状況

100GB～ / 実験
250～500 TiB / 年 増加
1TiB / 日 (100Mbits / sec) 増加

脳計測機器-発生データ(2000件/年)

3T-MRI　7T-MRI　MEG

PIMS + RIS Server

AIデータテストベッド整備

Large-scale memory (1.5TB) Server x 4

2 GPUs Server x 8

8 GPUs Server x 6

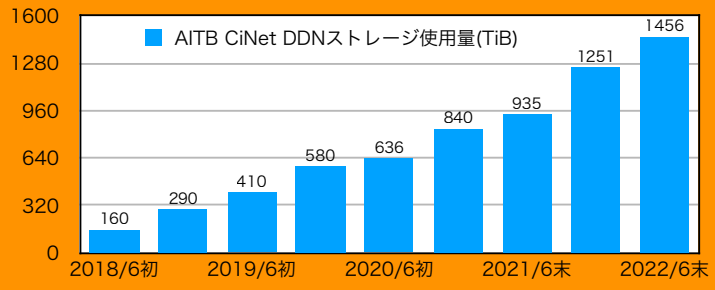
100Gbps IB network

DICOM Server　Flywheel

NFS Server

LDAP/AD Server

Storage (1.6 PiB)

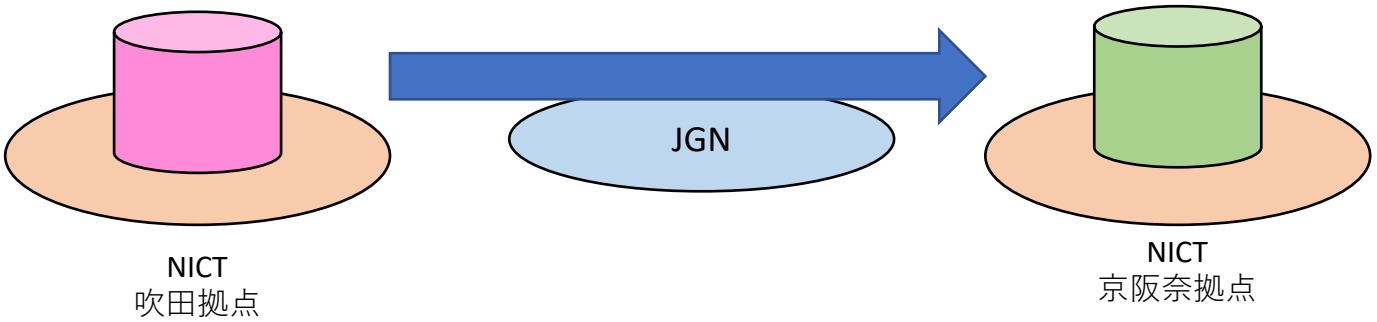


データバックアップ

- CiNet実験データのバックアップ
  - ファイルサーバ@吹田 （1.6 PB）の実験データ保全
  - 片方向コピーで保管でOK
  - NICT内他拠点（京阪奈）にコピー先を確保（DR?）

→ JGNを利用した大容量データコピーの実施

  - 2021年度に実施、本発表で共有したい内容
  - 最近の事例は見かけなかったので記録、フィードバックの議論のため

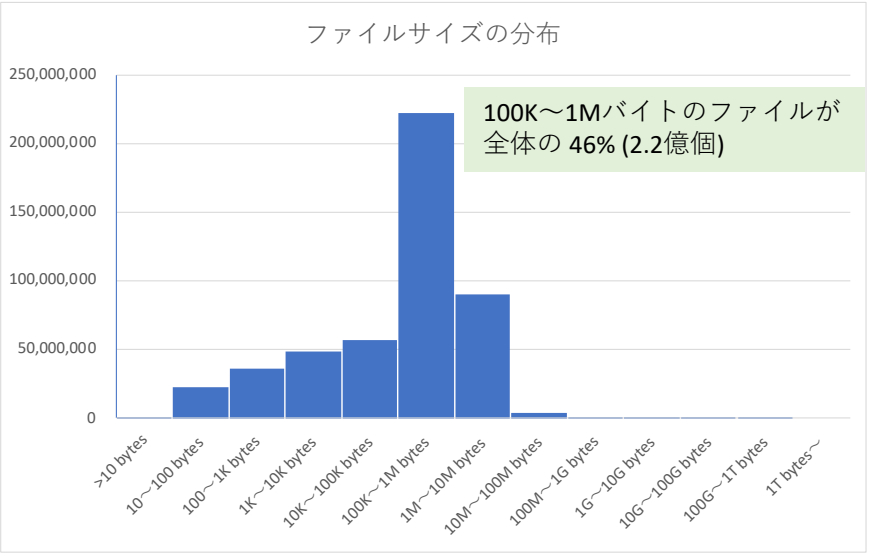


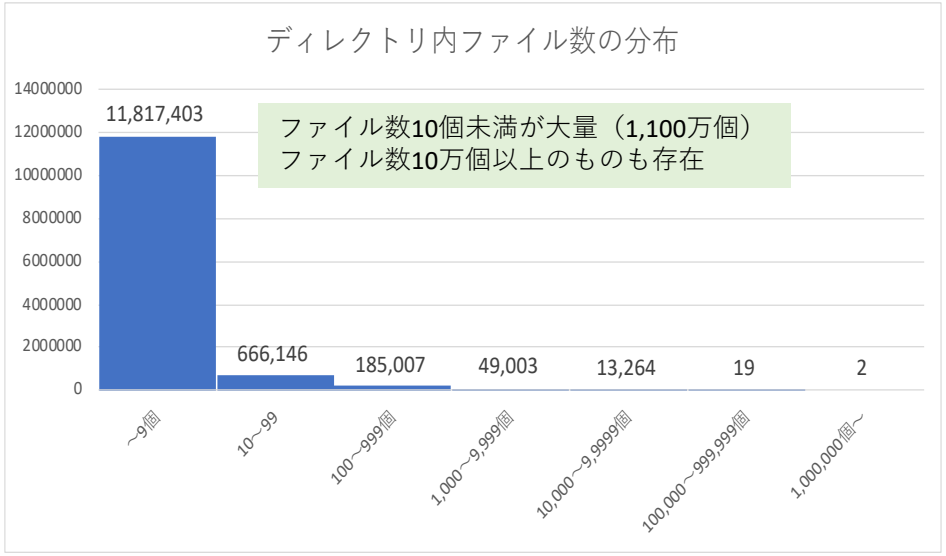
ファイル状況

FileSystem	1K-ブロック	使用	使用可	使用%	
data	1683895615488	1579303632896	104591982592	94%	/gpfs/data
(data	1.6P	1.5P	98T	94%	/gpfs/data)

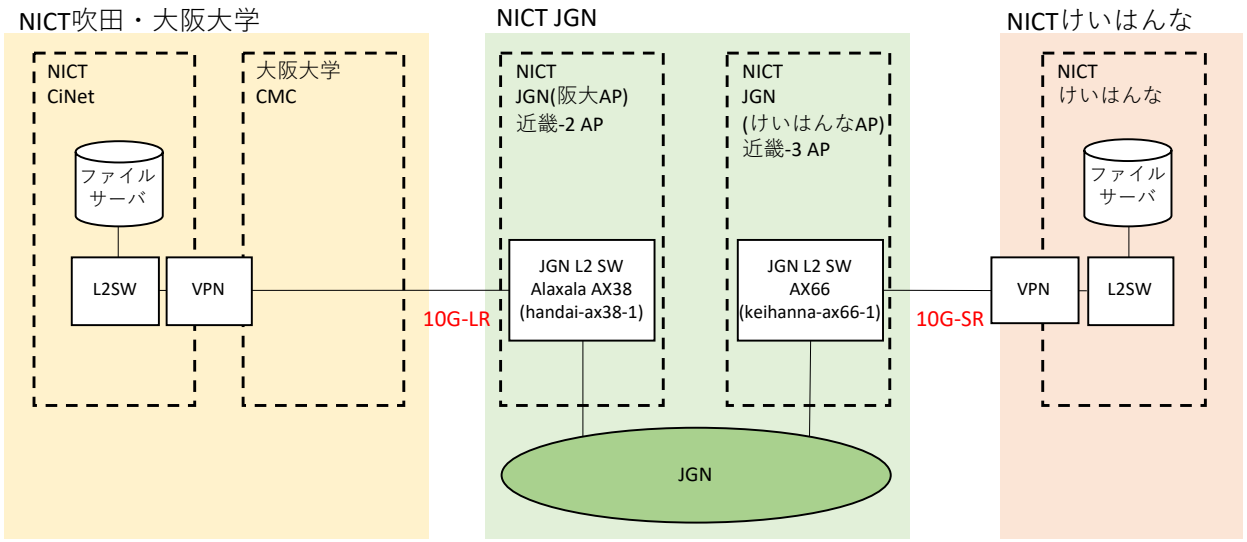
ファイル数	482,977,782 個
ディレクトリ数	12,730,843 個
ファイル容量	1,437,679,075,394,127 = 1.27 Pバイト

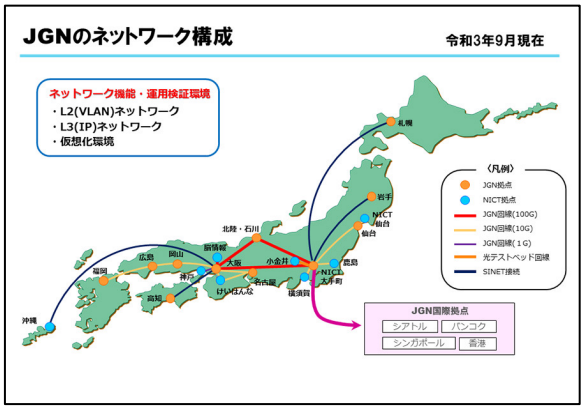
- 計測データ
- 画像データ
- 研究者のPCデータ など





ネットワーク状況





JGN経由
- 吹田～京阪奈 拠点間で10Gbps VLAN設定
- 大容量ストレージのバックアップ試験

- CentOS Linux 7 (Linux kernel 3.10.0), NFSv3利用
- JGNを経由した10Gbps VLANにて転送元と転送先を同一セグメント接続
- ネットワーク性能
  - 7.62 Gbps (iperf 2.0.13, TCP Windows size 128K Bytes, 5回平均)
  - RTTも1.5ms未満

```
100 packets transmitted, 100 received, 0% packet loss, time 99149ms
rtt min/avg/max/mdev = 1.290/1.359/1.435/0.054 ms
```

(1) ファイル読み書き性能

- ファイル読み書きの基本性能の確認
  - 理想的な最大性能の確認（これよりは早くならない）
  - ddを用いて直接的な読み書き性能の確認

  → 十分な性能 (最大 5Gbps 程度) が出ている
  → ファイルサイズが小さいとオーバーヘッド
  → OS & ネットワーク部分には問題なしと判断
    - cpでも1Gファイル x 10個 約20秒 = 約500MB/sec (=4Gbps)
- ローカル（データの読み込み）

データサイズ	1G bytes	100M bytes	10M bytes	1M bytes	100K bytes
転送速度	572 MB/sec	421 MB/sec	301 MB/sec	271 MB/sec	155 MB/sec

- NFS（遠隔への書き込み）

データサイズ	1G bytes	100M bytes	10M bytes	1M bytes	100K bytes
転送速度	580 MB/sec	535 MB/sec	303 MB/sec	69 MB/sec	10 MB/sec

```
# コマンド例
dd if=/LOCALNFS/dummy-1000M of=/dev/null  bs=1M count=1000
dd if=/dev/zero of=/REMOTENFS/dummy-1000M bs=1M count=1000
```

(2) rsyncでコピーのテスト（失敗）

- rsync試用
  - サンプルディレクトリ (300GB) にてrsync利用テスト
  - コピー元ローカルNFSからコピー先NFSへ
- 結果
  - 期待より長時間の3時間となった
    - 0.3TBを180分で転送 → 0.3 TB / 180 min = 約300Mbps
    - 1PBの転送には非現実的（400日長？）

```
# コマンド例
$ time rsync -avh /LOCALNFS/SAMPLE_DIR/ /REMOTENFS/TARGET_DIR/
```

rsyncコピーの遅さ原因調査 (1/2)

- コピー元データを生成して試験
  - テストパターンとして大量の小容量ファイル(1MB x 10000)、大容量ファイル(1GB) など
  - straceでrsync挙動を観測
- 結果
  - 小容量ファイルでは致命的な遅さ
  - 大容量ファイルでも転送速度は遅い（ddでのバルク転送と比較して1桁遅い）

```
$ strace -o rsync_write.txt -ttt -T rsync -ah LOCALSTORAGE /REMOTENFS/data
$ cat strace_rsync_write.txt | grep open
235.662097 open("files10000_1M/00000000", O_RDONLY) = 3 <0.000013>
237.530750 open("files10000_1M/00000001", O_RDONLY) = 3 <0.000021>
237.596222 open("files10000_1M/00000002", O_RDONLY) = 3 <0.000016>
237.719725 open("files10000_1M/00000003", O_RDONLY) = 3 <0.000019>
```

rsyncでのシーケンシャルなファイル読み込みの様子

```
$ cat strace_rsync_write.txt | grep open
315.410232 open("files10_1G/00", O_RDONLY)  = 3 <0.000018>
345.686952 open("files10_1G/01", O_RDONLY)  = 3 <0.000015>
376.190460 open("files10_1G/02", O_RDONLY)  = 3 <0.000012>
407.392757 open("files10_1G/03", O_RDONLY)  = 3 <0.000013>
```

1MBファイルのコピー
-ファイルコピーが遅い（1MBファイルのコピーに0.06〜1.5秒、5〜128Mbps）

1GBファイルのコピー
- 約30秒かかっている（266Mbps）、遅い

rsyncコピーの遅さ原因調査 (1/2)

- selectにて時折大きな処理遅延が発生
  - 小容量ファイルコピーでは致命的
  - cpによる小容量ファイルコピーでも再現、1MB x 10000個 (10GB)コピーにて6分50秒（=24 MB/sec = 192 Mbps）
- 読み書き処理 → 処理時間長
  - 26214バイトの読み込み (read)
  - 4092バイトに分割して書き込み (write)
    - write前にselect、「¥374¥17¥0¥7」パターンのwrite
    - 4KB writeだけなら1.6Gbps程度、読み込み＋前後処理の結果 250Mbps程度

→ rsync処理の限界？（NFSの限界？）

→ NFS経由でのrsync利用を断念

```
$ cat strace_rsync_write.txt | grep select | sort –k14
441.050028 select(5, NULL, [4], [4], {60, 0}) = 1 (out [4], left {59, 716175}) <0.283891>
376.191458 select(5, NULL, [4], [4], {60, 0}) = 1 (out [4], left {58, 852131}) <1.147950>
407.394002 select(5, NULL, [4], [4], {60, 0}) = 1 (out [4], left {58, 858102}) <1.141964>
468.456606 select(5, NULL, [4], [4], {60, 0}) = 1 (out [4], left {58, 861062}) <1.139013>
345.688570 select(5, NULL, [4], [4], {60, 0}) = 1 (out [4], left {58, 865743}) <1.134328>
```

```
$ cat strace_rsync_write.txt | grep write
315.410890 write(4, ..., 4092) = 4092 <0.000020>
315.411039 write(4,..., 4092) = 4092 <0.000019>
315.411187 write(4,..., 4092) = 4092 <0.000013>
315.411327 write(4,..., 4092) = 4092 <0.000014>
```

(3A) NFSを利用しないファイル転送実験
実験1：rsyncにてファイル転送

- rsync+SSHによるファイル転送
  - NFSを利用しない効果の確認、ツールの差の確認
- 結果
  - Rsync+SSH利用にて10Gバイトのファイル転送を試験
    - 約60秒で転送完了、ファイル数の影響無し
    - 10Gバイト / 64秒 = 1.25G bps （高速転送）

```
$ time rsync -au files10000_1M TARGET_PC:TARGET_DIR
real    1m4.778s → 1M x 10000個 = 10Gバイト 転送を64秒 （NFS経由だと 237 秒）
user    1m22.755s
sys     0m20.977s
$ time rsync -au files100_100M TARGET_PC:TARGET_DIR
real    1m3.327s → 100M x 100個 = 10Gバイト 転送を63秒 （NFS経由だと 1159 秒）
user    1m19.617s
sys     0m20.255s
```

(3B) NFSを利用しないファイル転送実験
実験2: rsync & tarにて実際のファイル転送

- サンプルデータ（300GB）を対象にして実験
  - 実際に近いデータによる検証
  - rsync, tarも使ってツール性能差も比較
- 結果
  - → 約300Gバイト 約60分 （NFS越しのときは約200分）
  - → ツール間で大きな性能差なし
    - 小ファイルも多く、約 600 Mbps の性能
- → この性能であれば、現実的な時間が見えてきた
  - 1 (PB) / 600 (Mbits / sec) = 166 days

```
$ time rsync -avh /LOCALNFS/SAMPLE_DIR TARGET_PC:TARGET_DIR
    551.60M 100%  135.72MB/s    0:00:03 (xfer#139328, to-check=3/139573)
sent 282.51G bytes  received 2.65M bytes  75.65M bytes/sec
total size is 282.47G  speedup is 1.00
2805.85user 781.14system 1:02:13elapsed 96%CPU (0avgtext+0avgdata 8292maxresident)k
0inputs+0outputs (0major+58910minor)pagefaults 0swaps
---
[ytel@braincs-mg01 project]$ time tar cvfp – SAMPLE_DIR | ssh TARGET_PC "cd TARGET_DIR && tar xvfp -"
14.32user 304.04system 59:36.13elapsed 8%CPU (0avgtext+0avgdata 1988maxresident)k
0inputs+0outputs (0major+1871minor)pagefaults 0swaps
```

(4) rsync+SSH並列実行

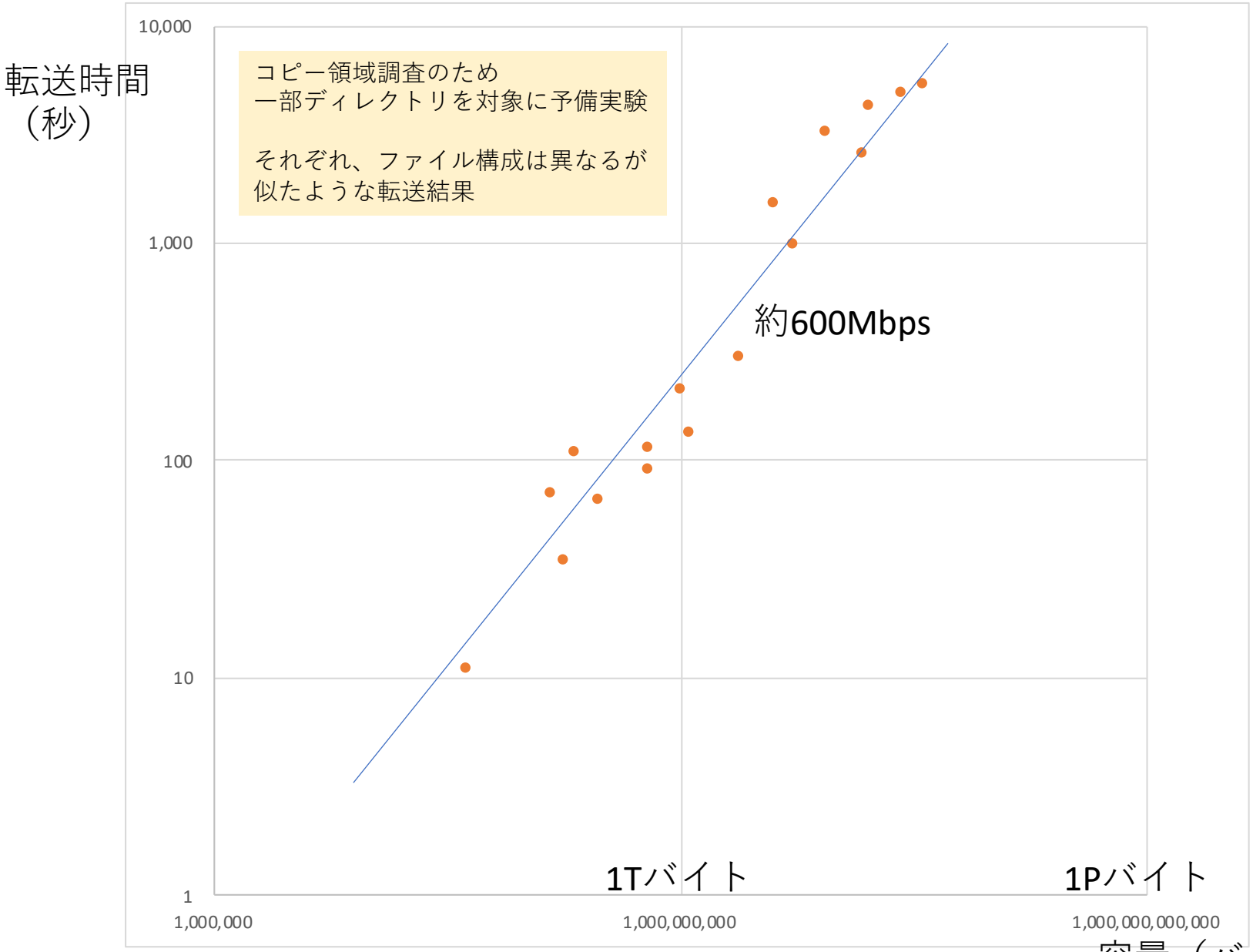
- rsync+SSHを複数同時起動
  - ベンチマーク中に発見
  - 約600Mbpsを維持したまま複数プロセス動作が可能
    - 8並列でも速度低下なく動作した（ネットワーク帯域には余裕）
- コピー領域の分割
  - 各プロジェクト向けディレクトリが存在
  - プロジェクトを集約して大きさが均等になるように設定
    - 領域1 335TB, 領域2 249TB, 領域3 175TB, 領域4 167TB
    - 領域1, 領域2のプロジェクトが巨大だったのでサイズを合わせて、4分割の領域に設定
- 結果
  - それぞれ600Mbps程度で並列コピーに成功
    - 最大の領域1が50日程度でコピー
  - 領域分割は困難で4並列に留まった
  - 圧縮すると低下、差分コピーも問題なく動作

ディレクトリ名	データ容量（Mバイト）	データ割合（%）	データ容量（Mバイト）
0001	334,605,374	36.16%	領域1 334,605,374 (36.16%)
0002	249,307,831	26.94%	領域2 249,307,831 (26.94%)
0003	67,109,542	7.25%	
0004	60,455,564	6.53%	領域3 174,941,683 (18.90%)
0005	47,376,576	5.12%	
0006	35,885,983	3.88%	
0007	26,298,222	2.84%	
0008	20,053,313	2.17%	
0009	16,485,150	1.78%	
0010	14,750,102	1.59%	
0011	14,607,860	1.58%	
0012	13,570,861	1.47%	
0013	8,465,054	0.91%	
0014	5,328,020	0.58%	
0015	4,020,992	0.43%	
0016	2,399,955	0.26%	
0017	1,154,158	0.12%	
0018	997,630	0.11%	
0019	624,617	0.07%	
0020	624,553	0.07%	
0021	381,256	0.04%	領域4 166,525,836,289 (18.00%)
0022	294,331	0.03%	
0023	208,356	0.02%	
0024	180,471	0.02%	
0025	144,664	0.02%	
0026	41,993	0.00%	
0027	8,246	0.00%	
0028	39	0.00%	
0029	1	0.00%	
0030	1	0.00%	
0031	0	0.00%	
0032	0	0.00%	
0033	0	0.00%	
0034	0	0.00%	
0035	0	0.00%	
合計	925,380,726		

- プロジェクト毎にディレクトリ
- プロジェクトを集約して領域分割
- 4領域に分割してコピー
- 領域1,2はそれぞれプロジェクト
  - 分割せずそのままコピー
  - これらがボトルネック
- 領域3,4は複数プロジェクト
  - 小さなプロジェクトを集約

- 領域分割はスクリプトで計算
  - 「ls -lR」結果を読み込み、領域分割を計算

各ディレクトリ容量と転送時間



作業と検討のまとめ

1. ローカル(NFS)とネットワーク部分の性能調査
   - 十分な性能（ネット 7.6Gbps, NFS 5Gbps）
2. rsync利用
   - 性能がでないケース、cpでも再現 （～300Mbps）
   - NFSの限界？
3. rsync+SSH利用
   - NFSを回避
   - 性能がやや回復（600Mbps～1.25Gbps程度）
4. rsync+SSHの並列実行
   - ディレクトリを集約してコピー
   - 1.85Gbps程度（ネットワークにはまだ余裕）

まとめ

- 結果
  - 10Gbpsリンク上での1PBコピーを約50日で完了 (1.85Gbps程度)
  - rsyncは有効、使い方には工夫が必要、差分コピーも動作
  - ボトルネックはネットワーク以外
- 疑問
  - ネットワーク越しのコピーのベストプラクティスは？
    - もっとよいやり方は？他の方はどうしている？（知識が古い可能性）
    - ネットワーク帯域を使い切るには？
    - rsync並列もディレクトリ構造に依存する
  - 他機関とのデータ交換や共有
    - 将来的にはMRIデータの交換などの検討
- その他
  - 次年度に20PBストレージ、10PB+10PB に分割してバックアップ
    - この遠隔バックアップも将来的な課題
  - GNU parallelsは不安定、parsyncfpという並列化ラッパは試してみたい