Catwalk Tutorial

東京大学
堀　敦史

ファイルステージング

- ファイルサーバ上のファイルを計算ノード間でコピー

  Stage-IN　　サーバ -> 計算ノード

  Stage-OUT　　計算ノード -> サーバ

- 短所
  - ジョブスクリプトに記述 – 間違えるとジョブ失敗
  - 遅い
    - 多くの場合，バックグラウンド処理なので速度の考慮なし
    - 計算時間にステージングの時間がプラス
- 長所
  - それほど強力なファイルサーバは必要ない

ファイルステージングの例

- JAXA さんの例

```
#!/bin/sh
#PBS ・・・
#F_FF10I=/data/input_file
#F_FF20O=/data/output_file
#F_FF30IO=/data/inoutput_file
./a.out
```

入力ファイル
出力ファイル
入出力ファイル

Catwalk

- 従来のステージングの欠点を改良
  - より並列ファイルシステムに近い使い易さ
  - より高性能なファイルステージング
- 並列ファイルシステムの欠点も改良
  - 強力なファイルサーバの必要性
  - システム依存性をなくし，可搬性を向上
- 協調的なシステム
  - 既存の分散／並列ファイルシステムとの共存，使い分けが可能

Catwalk の概要

- ふたつの Catwalk
  - Catwalk
    - オンデマンドファイルステージング
    - ステージングの記述が不要なので、記述を間違えない
    - NFS と比肩できる性能
  - Catwalk-ROMIO
    - MPI-IO に特化した Catwalk
    - 並列ファイルシステムと比肩できる(or それ以上)の性能
- １台のファイルサーバのみをサポート

Catwalk の特長

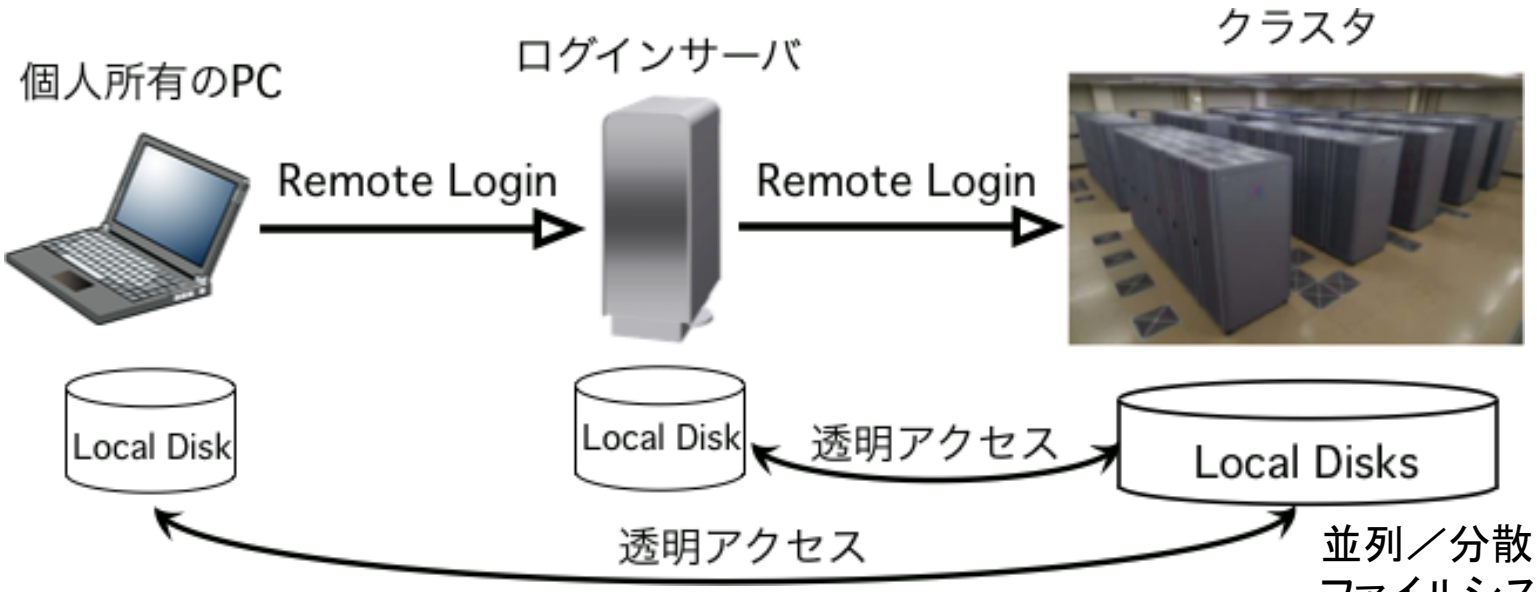
- インストールに root 権限不要
  - ビルドして PATH に登録するだけ
  - 独自の設定ファイルなし
- 特別なハードは不要
  - ファイルサーバ　　普通のディスクとネットワーク
  - 計算ノード　　普通のディスクとネットワーク
  - ネットワーク　　Ethernet (GbE) で十分
- 対象となるファイルは普通のファイル
  - 変換やコピーの必要なし

Catwalk のモデル

- ファイルサーバにあるファイルを計算ノードからアクセスできる
  - プログラムからは Catwalk の存在は見えない
- 必要に応じて計算ノードのディスクにキャッシュを生成する
  - キャッシュはプログラムの終了時に自動的に削除
- 計算ノードに書込まれたファイルはファイル最終的にファイルサーバ上に書込まれる
  - MPI-IO で生成しても通常のファイルとして生成

Catwalk の使い方モデル

- 個人が使っている PC あるいはログインサーバ上のファイルを，クラスタから直接アクセス（しているように見せかける）
- 陽にクラスタファイルシステムにコピー不要
- Catwalk の性能は並列／分散ファイルシステムに負けない
  （どっちが良いかはケースバイケース）



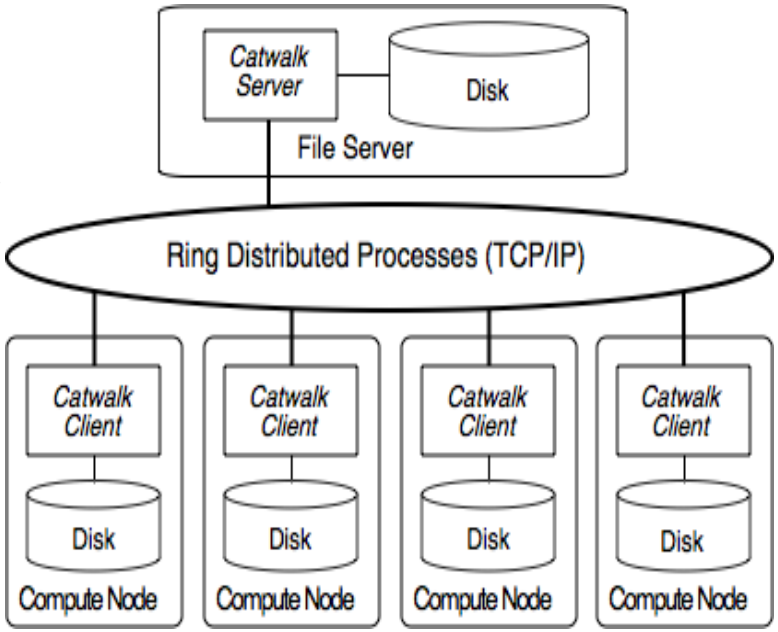
Catwalk の内部

リング分散プロセス構造

並列アクセスを出来るだけ逐次アクセスに

→ シークを減らして高速化

- Stage-IN
  - 対象ファイルを全てローカルディスクにコピー
- Stage-OUT
  - 逐次全体コピー
- MPI-IO Read
  - Stage-IN と同じ
- MPI-IO Write
  - 並列書込要求を逐次書込に変換



アジェンダ

- ふたつの Catwalk
  - 逐次／並列プログラム　　Catwalk
  - 並列プログラム（MPI-IO）　　Catwalk-ROMIO
- それぞれについて
  - 使い方の詳細
  - 現版の制限事項
- 性能

How to Use Catwalk

Catwalk の使い方 – 基本

% catwalk -nh 2 mpirun -np 4 catwalk a.out

nh : ノード(ホスト)数 プロセス数ではない！

- Stage-IN : a.out 内で read-open したファイルが a.out のカレントディレクトリにコピー
- Stage-OUT: a.out がwrite-open したファイルがプログラム終了後にファイルサーバにコピー

計算ノードの対象ファイル

- Stage-IN, Stage-OUT 共通
  - ファイル名は "/" なしでなければならない
  - プログラム中でカレントディレクトリ変更可能
  - カレントが NFS 等の場合の動作保証なし
- Stage-IN
  - カレントディレクトリに同名ファイルがない事
- Stage-OUT
  - カレントディレクトリのファイルをサーバへ
  - いったん作ったファイルを消した場合，対象外

サーバの対象ファイル

- Stage-IN

  % catwalk -nh 32 -path *A:B:C* mpirun catwalk a.out

  a.out で read-open したファイル名をサーバのディレクトリ *A,B,C* の順で探し，あればコピー.

- Stage-OUT

  % catwalk -path *X:Y:Z* mpirun catwalk a.out

  % catwalk -dir *X* mpirun catwalk a.out

  a.out で write-open したファイルをサーバ上のディレクトリ *X* にコピー(移動)

Catwalk の使用例- read

% cat ＞ *DIR*/foo.dat

適当なファイル foo.dat がサーバにあるとして

% mpirun −np 1 wc *DIR*/foo.dat

wc: *DIR*/foo.dat: No such file or directory

計算ノードには foo.dat というファイルが無くても

% catwalk −nh 1 −dir *DIR* mpirun −np 1 catwalk wc foo.dat

Catwalk を使うことで計算ノード上でサーバ上のファイル *DIR*/foo.dat を読むことができる

Catwalk の使用例 - write

```
% catwalk –nh 2 mpirun –np 2 catwalk touch X
% ls
 X@comp0              X@comp1
% catwalk –nh 2 mpirun –np 2 catwalk touch X
% ls
X@comp0    X@comp1    X@comp0#1    X@comp1#1
% catwalk –nh 2 mpirun –np 2 catwalk touch X
% ls
X@comp0    X@comp1    X@comp0#1    X@comp1#1
X@comp0#2    X@comp1#2
%
```

Catwalk の制限事項

- Dynamic Link Program
- サポートしている glibc 関数

  creat(2), open(2), fopen(3), stat(2), access(2), exec() 系列
- サポートしているコンパイラ

  GNU（g77は除く），Intel，PGI 他
- セキュリティな環境で使うことが前提

How to Use Catwalk-ROMIO

Catwalk-ROMIO 基本操作

% catwalk -mpi mpirun catwalk a.out

a.out で MPI_File_open("catwalk:/xxx/yyy") とするとサーバ上の /xxx/yyy がアクセスされる

% catwalk -mpi mpirun catwalk -dir *ZZZ* a.out

Catwalk-ROMIO の一時ファイルを計算ノードのディレクトリ *ZZZ* に生成する

Catwalk-ROMIO 応用編

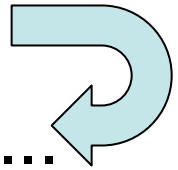
- ssh-agent 風な使い方

  必ずセキュアな環境で使う事！

  % catwalk –mpi

  export XXX=YYY ...

  % export XXX=YYY ...



メッセージが表示される

それをコピペする

  % mpirun catwalk a.out

  ...

  % catwalk –kill　　デーモンプロセスを kill

Catwalk-ROMIO SSH (1)

server% catwalk –mpi –SSH comp32

Catwalk 環境と SSH port forwarding を設定し

comp32 に ssh でログイン

comp32% mpirun catwalk a.out

server 上のファイルを MPI-IO でアクセス.

複数ホップも OK

comp32% catwalk –SSH comp78

comp78% mpirun catwalk a.out

Catwalk-ROMIO SSH (2)

server% catwalk –mpi bash

Catwalk 環境下の bash を生成

セキュアな環境でのみ使う事！

server% catwalk –SSH comp345

Catwalk 環境を引き継いで comp345 へ

comp345% mpirun catwalk a.out

SSH を経由して server のファイルを MPI-IO

共通オプション

- Catwalk 及び Catwalk-ROMIO 共通オプション

ネットワークインターフェイスの指定

% catwalk -net <IF名> ....

IF 名は /sbin/ifconfig で表示されるネットワークインターフェイスの名前.

例えば eth0, myri0, ib0, ...

Catwalk-ROMIO の制限

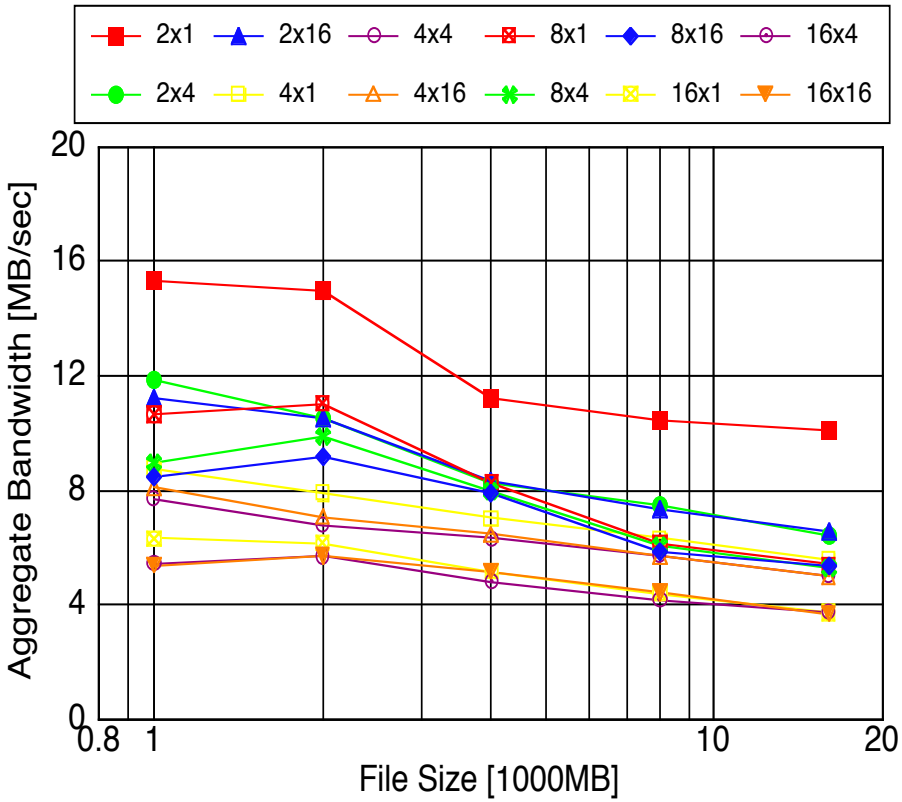
- Catwalk（非ROMIO）と同時に動かない
- SCore 付属の MPICH でしか動かない
- MPI-IO の atomic モードには対応していない
- MPI-IO を使った1プロセス1ファイルのアクセス性能は低い

性能比較

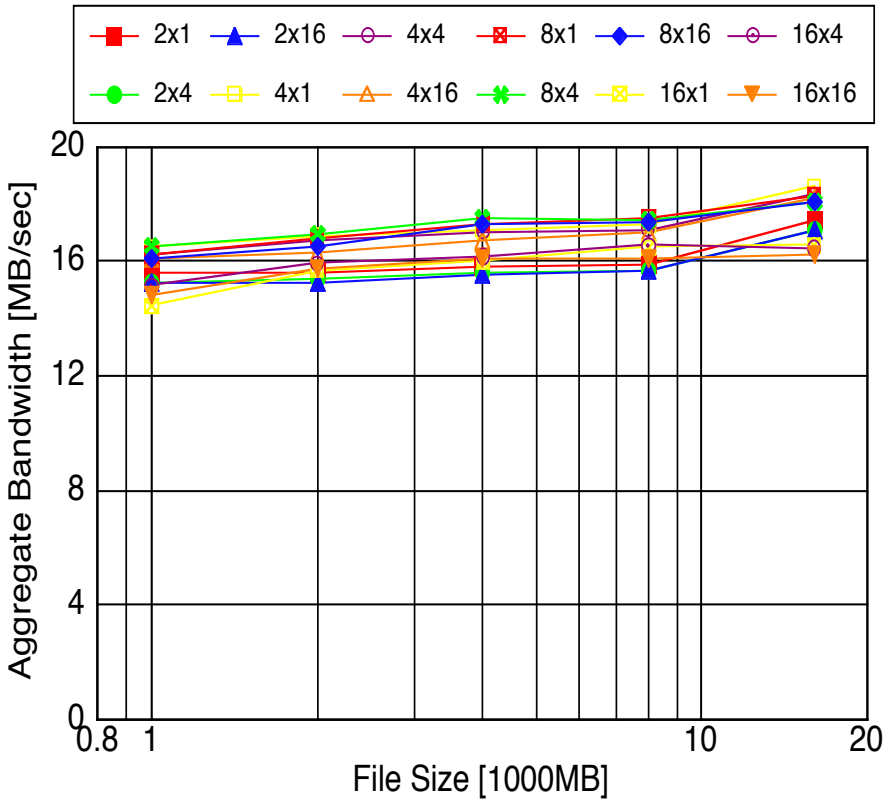
Catwalk vs. NFS (1)

- ひとつのファイルを各プロセスが 1/N 読む

NFS



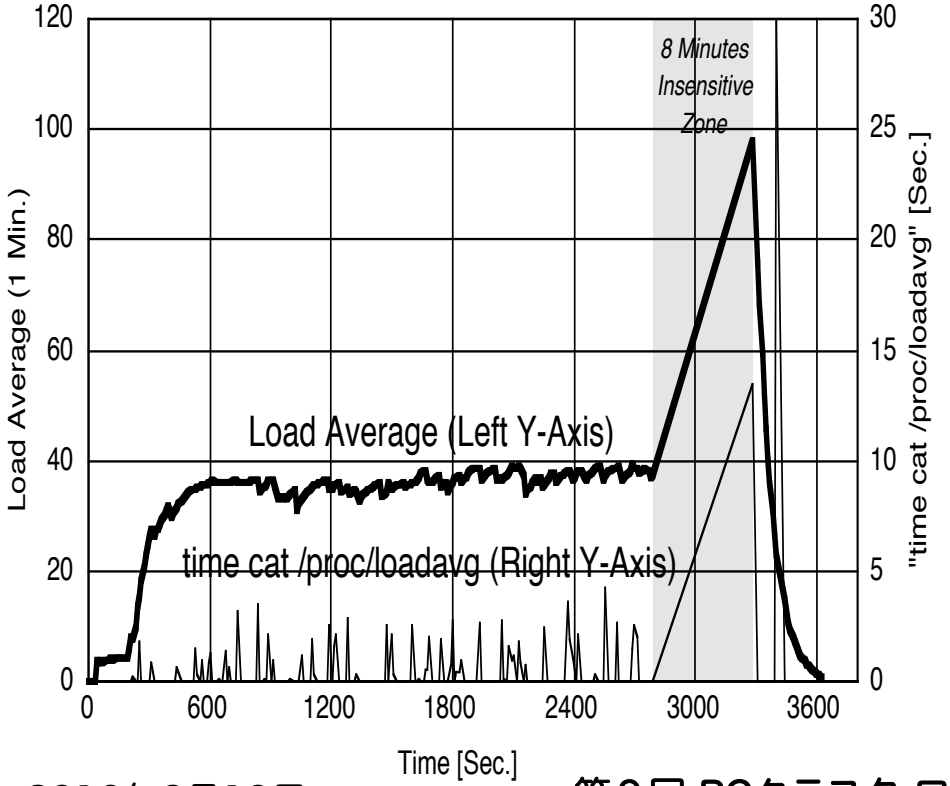
Catwalk



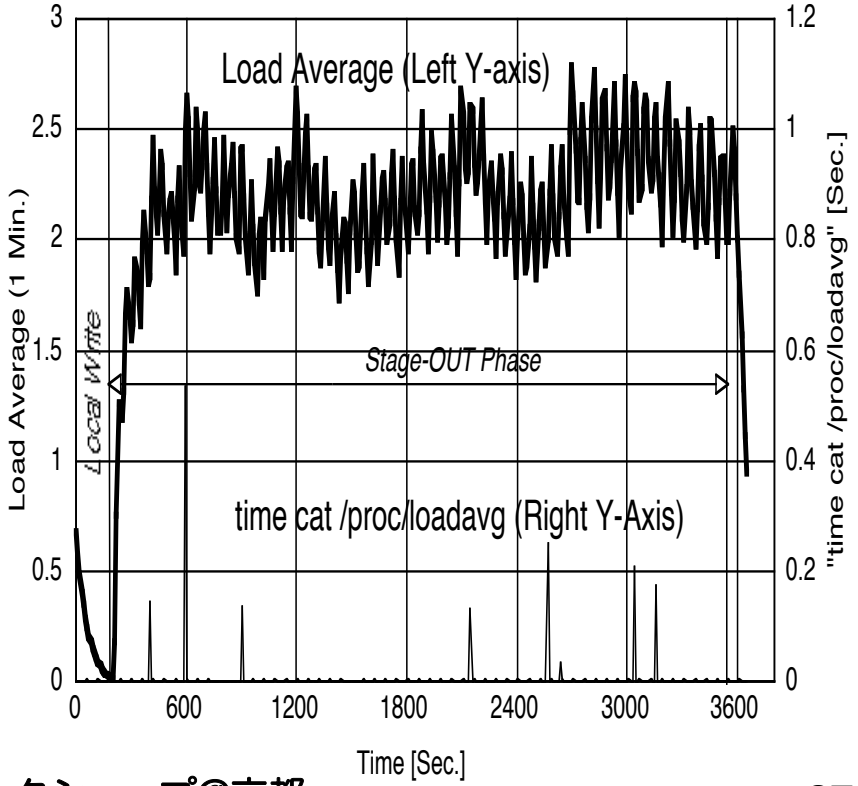
Catwalk vs. NFS (2)

- 4x16 の各プロセスが 1GB のファイルを書込む際のサーバの負荷

NFS

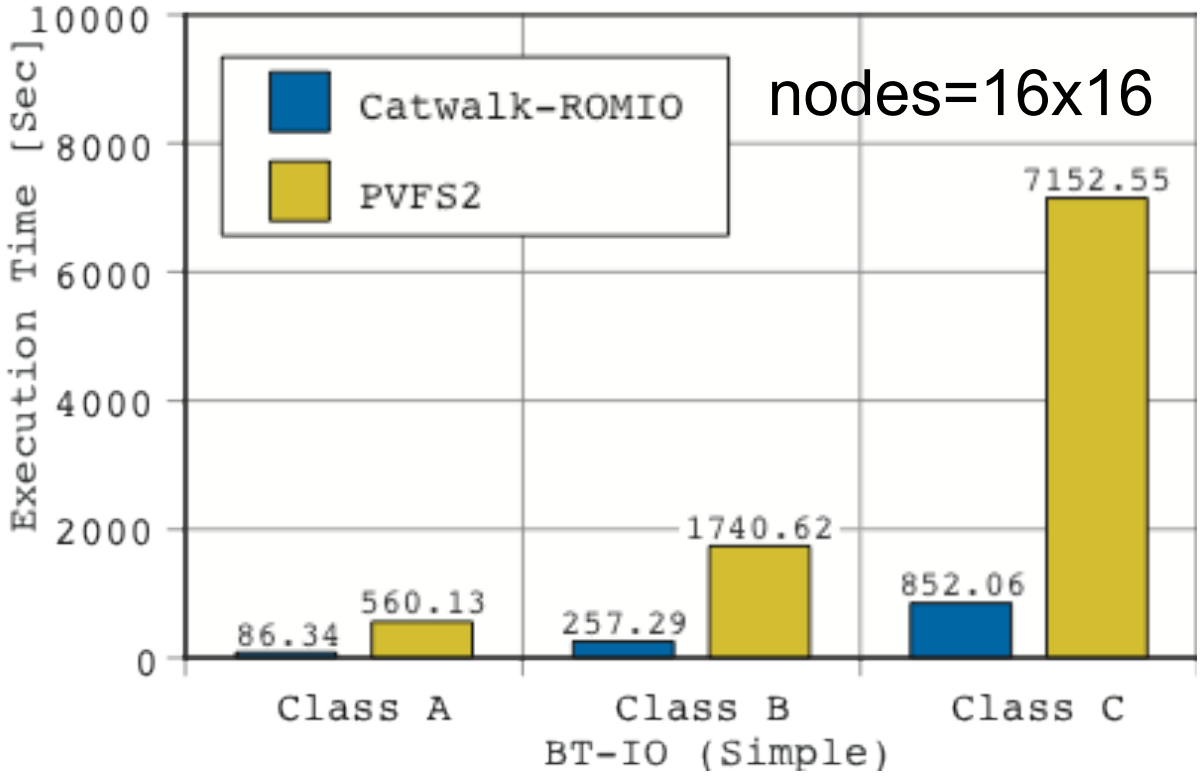


Catwalk

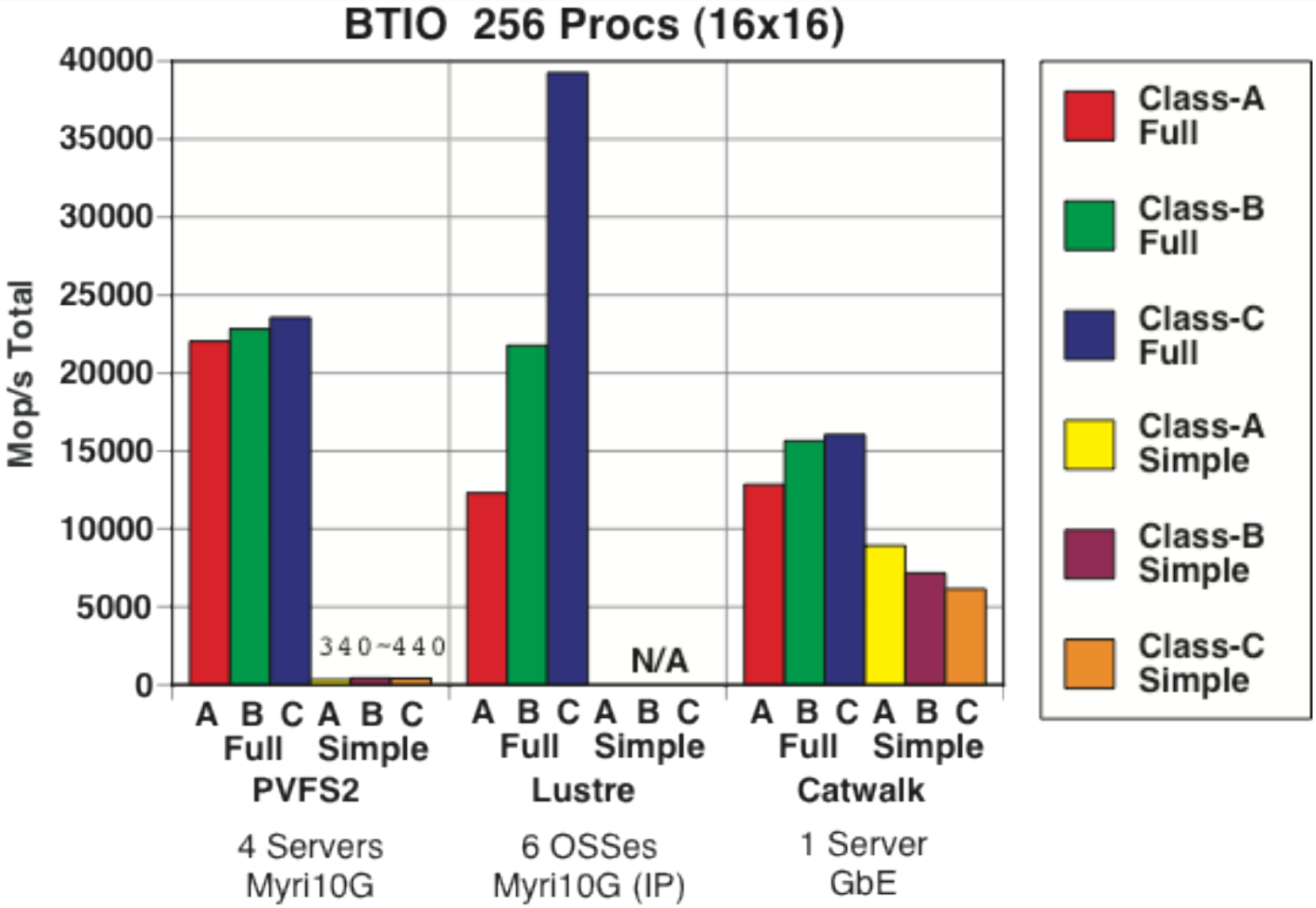


Catwalk-ROMIO vs. PVFS2

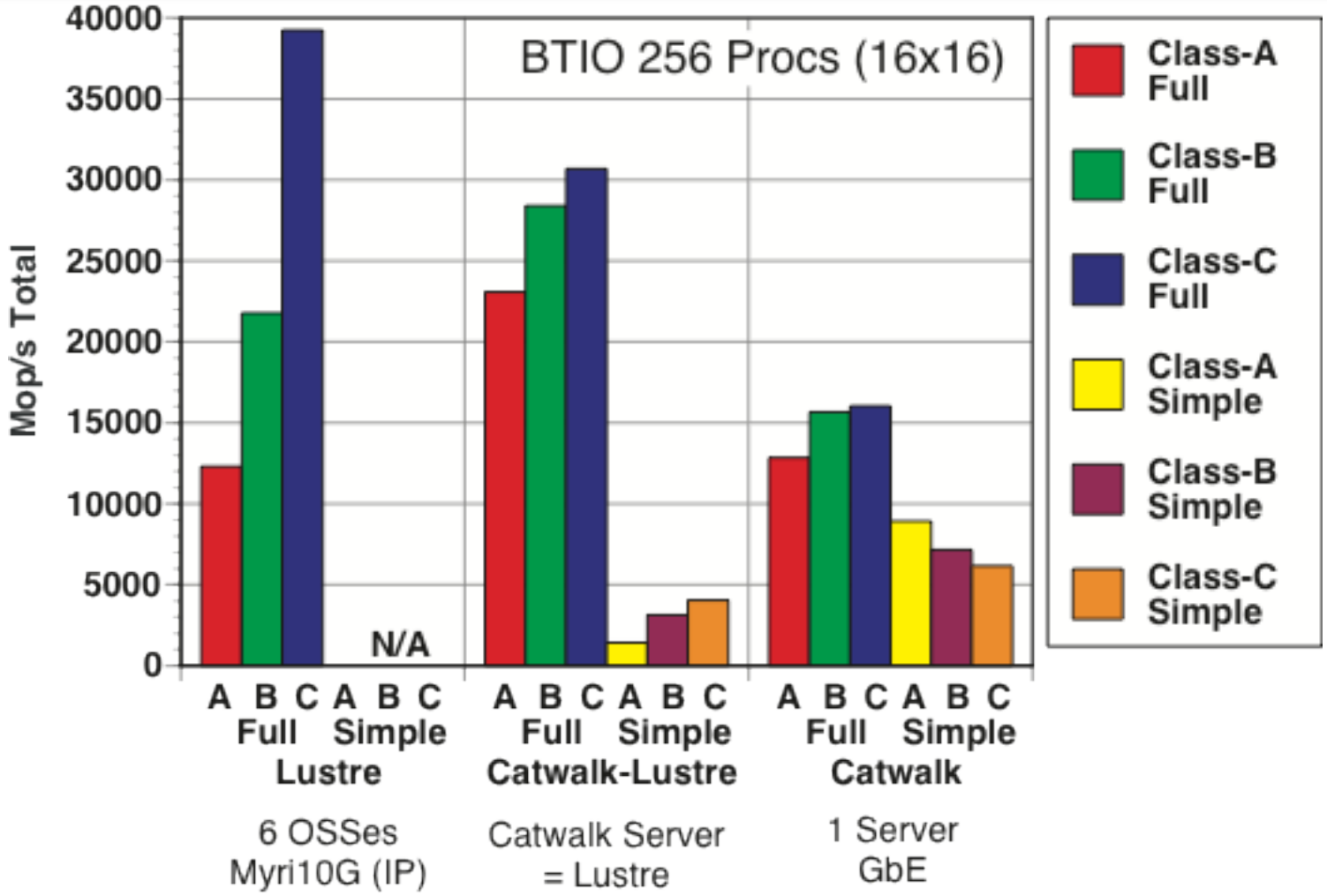
- Catwalk-ROMIO
  サーバ1台
  Ethernet (1 Gb/s)
- PVFS2
  サーバ4台
  Myri10G (10 Gb/s)



BT-IO ベンチマーク


BTIO 256 Procs (16x16)
Mop/s Total
40000
35000
30000
25000
20000
15000
10000
5000
0
340~440
N/A
A B C A B C
Full Simple
PVFS2
Lustre
Catwalk
4 Servers
Myri10G
6 OSSes
Myri10G (IP)
1 Server
GbE
Class-A Full
Class-B Full
Class-C Full
Class-A Simple
Class-B Simple
Class-C Simple

Catwalk-ROMIO and Lustre



Q&A