

# SMBDirectBMTまとめ

※今回使用したSTREAM V11及びSCRYU/Tetra V11はβ版ですので、正式リリースとは仕様が異なる可能性がございます。

※SCRYU/Tetraで用いている機能「MSPARIO」は現在未公開の機能となっております。

2013年08月

株式会社ソフトウェアクレイドル  
技術部  
阿部



# はじめに

- 昨今では解析データが大規模化し、かつ非定常な現象を解析することが増えてきているが、STREAMもSCRYU/Tetraも、このような解析では計算ノードからの図化ファイル(FLDファイル)の出力や、POST処理のためのFLDファイルの読み込み等、ファイルI/Oに要する時間が無視できないほど増加している。
  - このような状況を改善するためには計算ノードからファイルサーバーへ高速なI/Oを行える共有ストレージシステムや、POST処理のためにファイルサーバーからFLDファイルを効率良く読み込むためのシステムの構築が必要となる。
- WindowsServer2012で採用されたSMB3.0ではInfinibandネットワーク上でSMBdirectと呼ばれる機能を用い、高速で安定したストレージ環境を構築できる。このSMBdirectを用いてSTREAM,SCRYU/Tetraを用いてベンチマークを実施した。



# ベンチマーク実施環境1

## ● FILESERVER

- MACHINE: Dell PowerEdge R720
- OS: Windows Server 2012 + HPC Pack 2012
- CPU: Intel Xeon [E5-2665@2.40GHz](#)  
2CPU16core(HTTon-32core)
- MEM: DDR3-1600MHz 8GB\*12
- NIC: Intel GbE I350c rNDC , Intel GbE I350-T4
- IB : Mellanox ConnectX-3  
(FW:2.11.500,Driver:4.401.4223.0)
- SSD: DELL\_P320h-MTFDGAL350SAH \* 4台



# ベンチマーク実施環境2

- **COMPUTE-NODE (4nodes)**

- MACHINE: Dell PowerEdge R620
- OS: Windows Server 2012 + HPC Pack 2012
- CPU: Intel Xeon E5-2690@2.90GHz,2CPU16core(HTToff)
- MEM: DDR3-1600MHz 8GB\*12
- NIC: Intel GbE I350c rNDC , Intel GbE I350-T4
- IBHCA : Mellanox ConnectX-3  
(FW:2.11.500,Driver:4.401.4223.0)



# ベンチマーク実施環境3

- **InfiniBandSwitch: Mellanox SX6036**
- **ソフトウェア**
  - SCRYU/Tetra V11 beta
    - SCTsolver\_Bx64net.exe : 6111.20300.20130730
    - sctsol\_Dx64net.exe : 6511.22301.20130725
  - STREAM V11 beta
    - Stsolver\_BX64net.exe : 1111.20300.20130730
    - sctsol\_Dx64net.exe : 1511.22300.20130726
  - その他
    - MPI:MSMPI , SMB : v3.0



# ベンチマーク実施環境4

- MPIの通信は全てInfiniband(以下IB)経由で行った。
- ファイルI/OはIB(SMBdirect)経由と1GbE経由で比較した。
- STREAMではV10までのFLD出力方式と、V11から採用の新しいFLD出力方式でも比較を行った。(詳細は後述)
- SCRYU/TetraではPOSIXIOでのFLD出力方式と、MSPARIOでのFLD出力方式でも比較を行った。(詳細は後述)



# STREAM: FLDファイル出力方式について

- **V10までのFLDファイル出力方式(以降V10方式とする)**

1. 各MPIプロセスが、自分が担当した計算部分のデータをストレージに書き込む。
2. 計算終了時、各MPIプロセスから出力されたデータを読み込み、各MPIプロセスが担当した部分のFLDファイルを作成。作成したFLDファイルをストレージへ書き込む。
3. 解析終了後、gather処理により各MPIプロセスが作成したFLDファイルが読み込まれ、1つのFLDファイルとしてストレージへ書き込まれる。

- **V11からのFLDファイル出力方式(以降V11方式とする)**

1. 各MPIプロセスが、自分が担当した計算部分のデータをMPI通信を用いてRANK0へ集める。(ストレージは介さない)
2. RANK0がFLDファイルを作成し、作成したFLDファイルをストレージへ書き込む。(ストレージへのアクセスはFLDファイル書き込みの1度だけ)

- **ストレージへのアクセスが少ないためV11方式のほうが高速なことが期待できる。**



# STREAM: V10方式とV11方式の比較

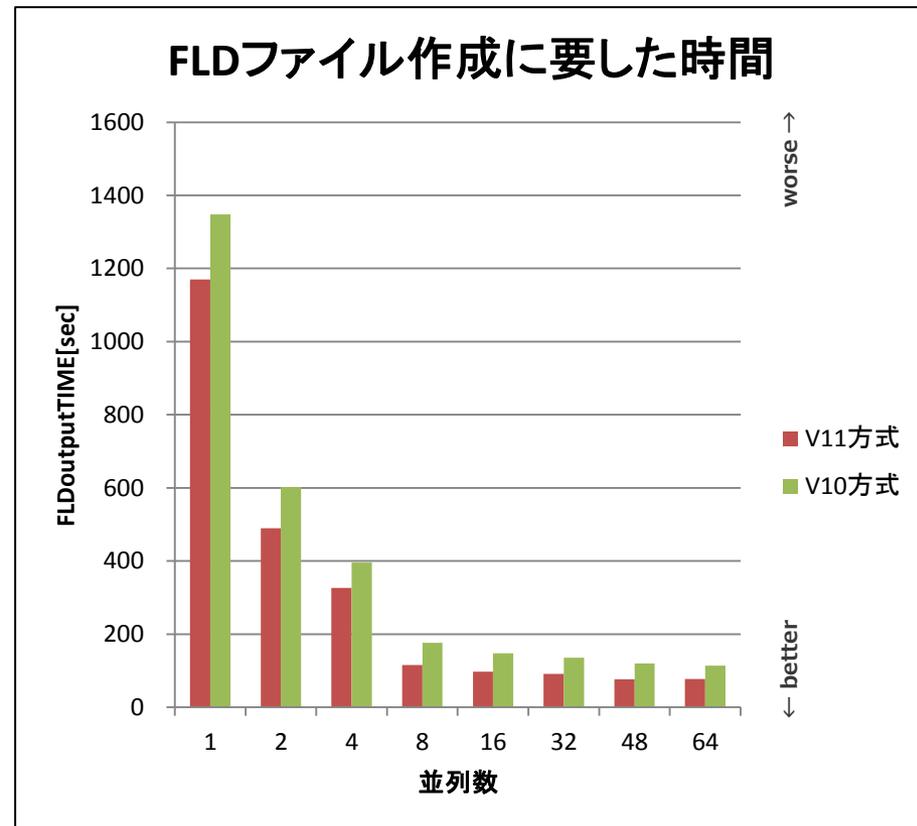
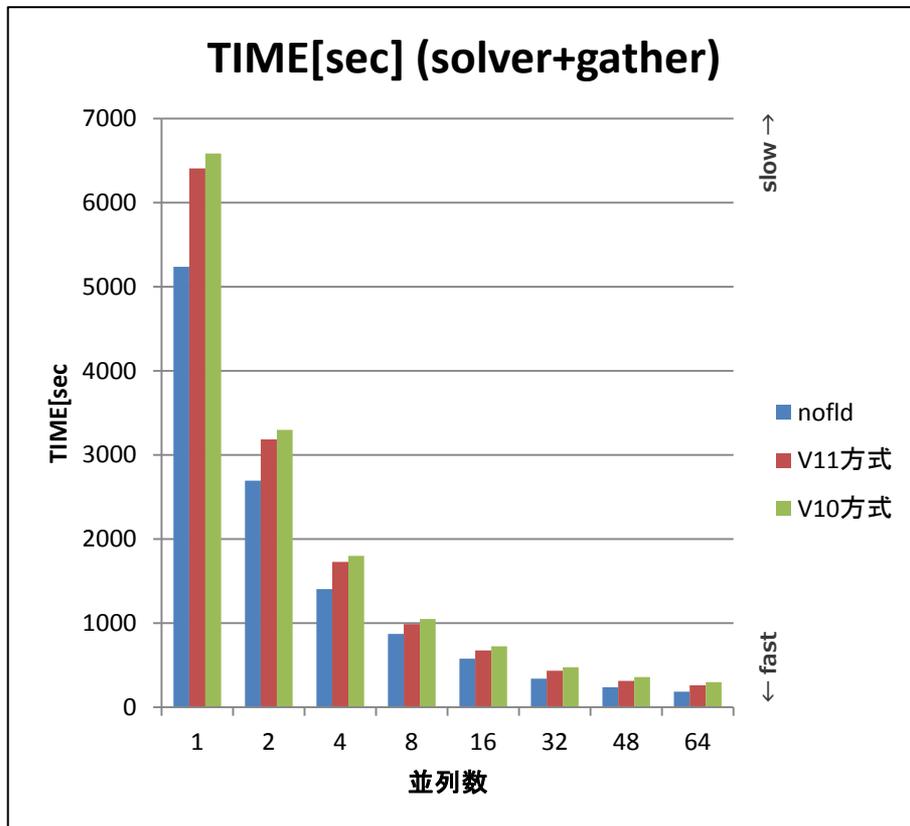
## ● Input:Tsunami (10cycle)

- IB(SMBdirect)経由でストレージへFLDファイルを書き込む。
- TIME = 解析開始からFLDファイル出力完了までの時間
- nofld: 同じ解析でFLDファイル出力無しの場合
- FLDoutputTIME = (各方式のTIME) - (nofldのTIME)  
= FLDファイルの出力に要した時間

| 並列数 | TIME[sec]<br>(solver+gather) |        |        | FLDoutputTIME[sec] |        |
|-----|------------------------------|--------|--------|--------------------|--------|
|     | nofld                        | V11方式  | V10方式  | V11方式              | V10方式  |
| 1   | 5235.4                       | 6405.0 | 6583.3 | 1169.7             | 1347.9 |
| 2   | 2694.4                       | 3184.3 | 3296.9 | 489.9              | 602.4  |
| 4   | 1403.7                       | 1730.1 | 1799.3 | 326.5              | 395.7  |
| 8   | 873.1                        | 989.2  | 1049.3 | 116.1              | 176.2  |
| 16  | 579.5                        | 676.8  | 727.2  | 97.3               | 147.7  |
| 32  | 341.8                        | 433.3  | 477.4  | 91.5               | 135.7  |
| 48  | 238.9                        | 315.7  | 359.1  | 76.8               | 120.2  |
| 64  | 185.0                        | 262.2  | 298.6  | 77.2               | 113.6  |



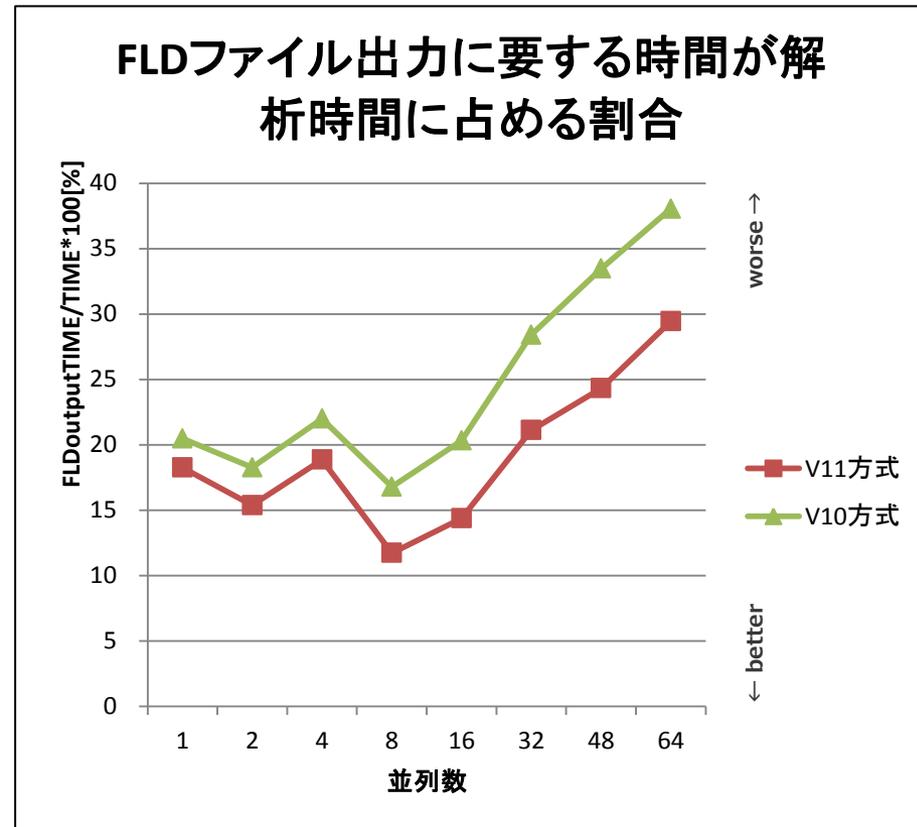
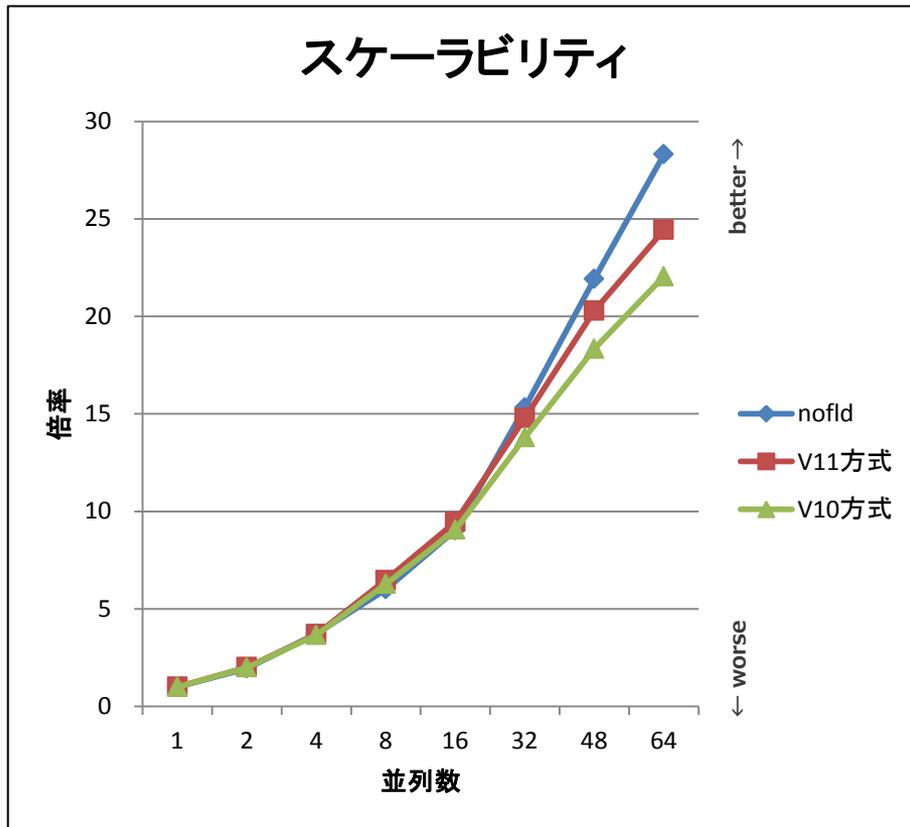
# STREAM: V10方式とV11方式の比較



- V11方式のほうがFLDファイル完成までに必要な時間を短縮できている。



# STREAM: V10方式とV11方式の比較



- スケーラビリティもV11方式を用いたほうが良い。



# STREAM: 1GbEとIB(SMBdirect)の比較

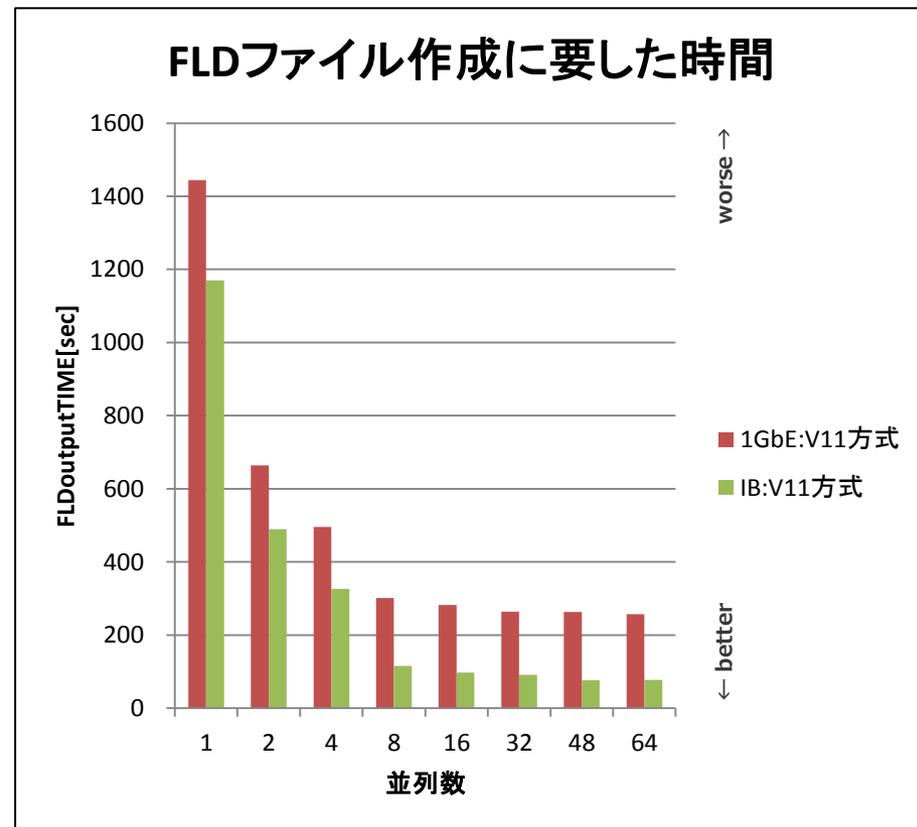
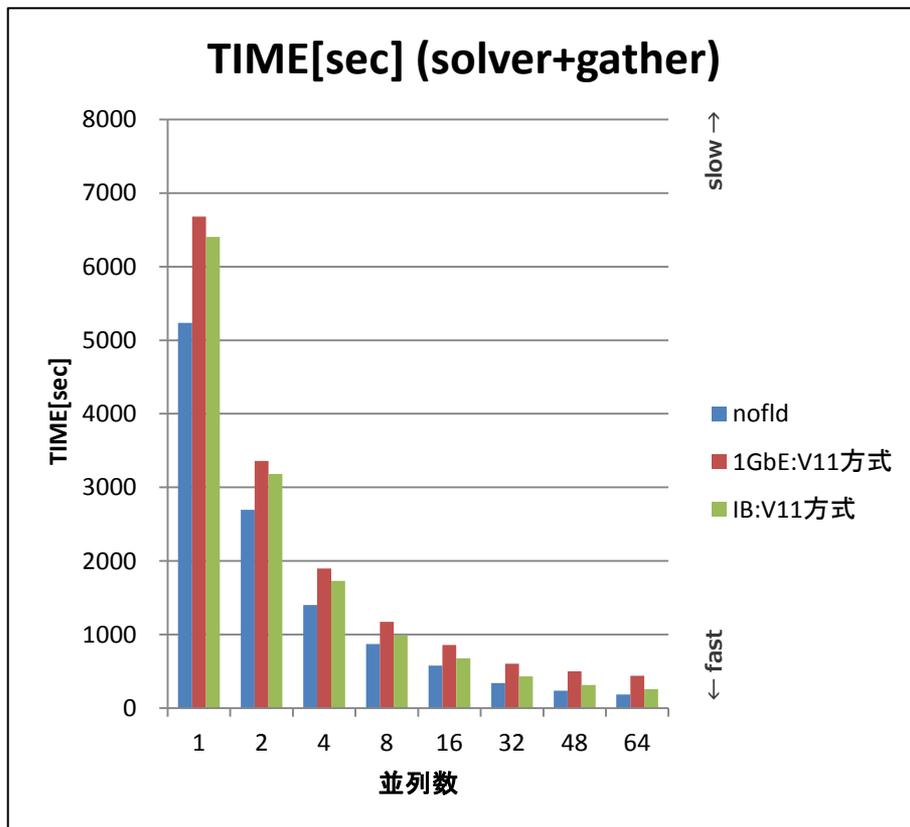
## ● Input:Tsunami (10cycle)

- IB(SMBdirect)経由と1GbE経由でストレージへFLDファイルを書き込む。
- TIME = 解析開始からFLDファイル出力完了までの時間
- nofld: 同じ解析でFLDファイル出力無しの場合
- FLDoutputTIME = (各方式のTIME) - (nofldのTIME)  
= FLDファイルの出力に要した時間

| 並列数 | TIME[sec] |            |          | FLDoutputTIME[sec] |          |
|-----|-----------|------------|----------|--------------------|----------|
|     | nofld     | 1GbE:V11方式 | IB:V11方式 | 1GbE:V11方式         | IB:V11方式 |
| 1   | 5235.4    | 6679.4     | 6405.0   | 1444.0             | 1169.7   |
| 2   | 2694.4    | 3358.6     | 3184.3   | 664.2              | 489.9    |
| 4   | 1403.7    | 1899.8     | 1730.1   | 496.1              | 326.5    |
| 8   | 873.1     | 1174.9     | 989.2    | 301.7              | 116.1    |
| 16  | 579.5     | 861.4      | 676.8    | 281.9              | 97.3     |
| 32  | 341.8     | 606.1      | 433.3    | 264.3              | 91.5     |
| 48  | 238.9     | 502.3      | 315.7    | 263.4              | 76.8     |
| 64  | 185.0     | 441.8      | 262.2    | 256.8              | 77.2     |



# STREAM: 1GbEとIB(SMBdirect)の比較

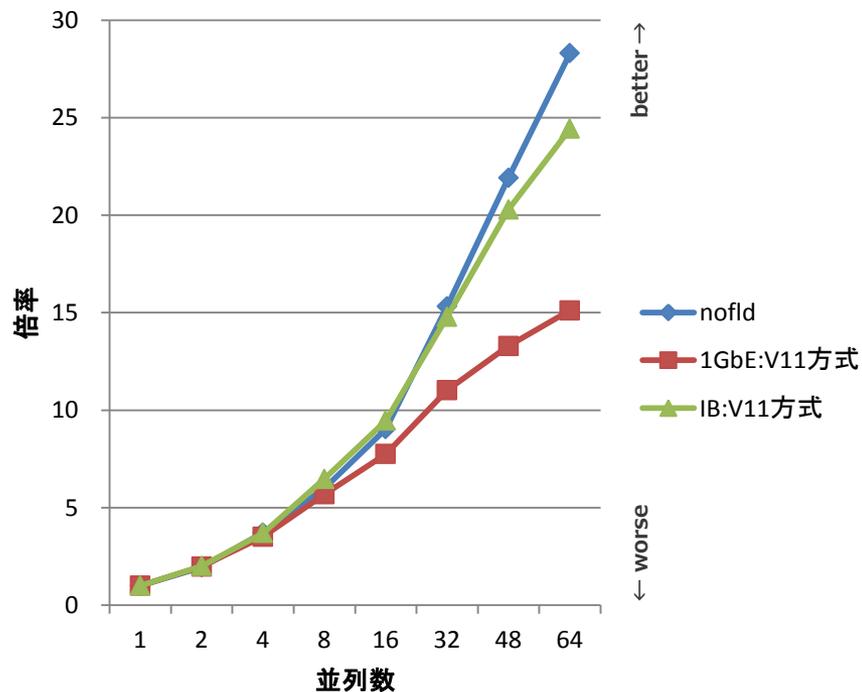


- 1GbEではFLDファイルをストレージへ書き込む際に時間がかかっていると予想される。

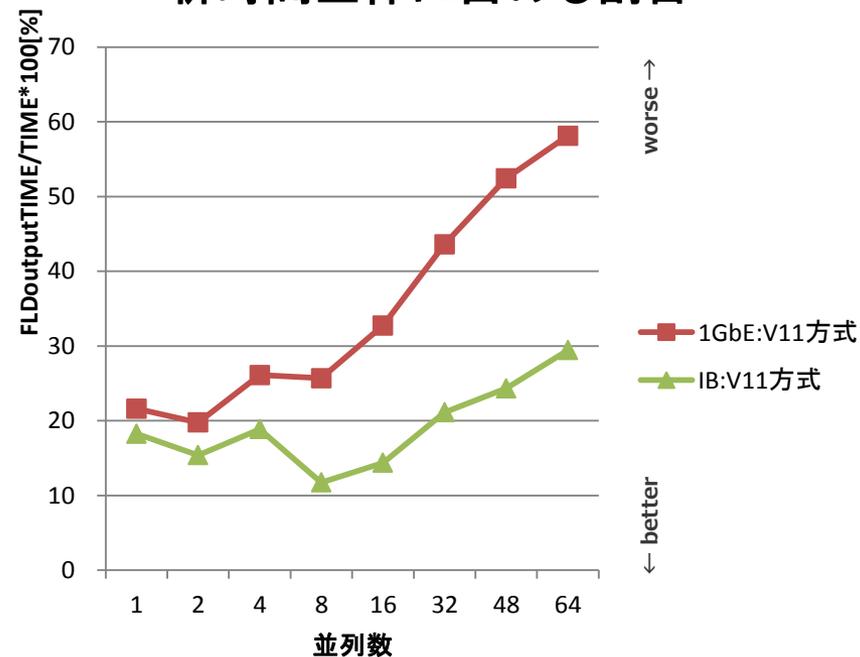


# STREAM: 1GbEとIB(SMBdirect)の比較

スケーラビリティ



FLDファイル出力に要する時間が解析時間全体に占める割合



- IB(SMBdirect)ではFLDファイル出力無しの場合に近いスケーラビリティを見せる



# SCRYU/Tetra: FLDファイル出力方式について

- **POSIXIO(STREAMのV11方式と同じ)**

1. 各MPIプロセスが、自分が担当した計算部分のデータをMPI通信を用いてRANK0へ集める。(ストレージは介さない)
2. RANK0がFLDファイルをストレージへ書き込む。

- **MSPARIO**

1. MPI I/O関数を用いて、各MPIプロセスが平行してストレージへFLDファイルを書き込む。(RANK0へは集めない。)



# SCRYU/Tetra: POSIXIO vs MSPARIO

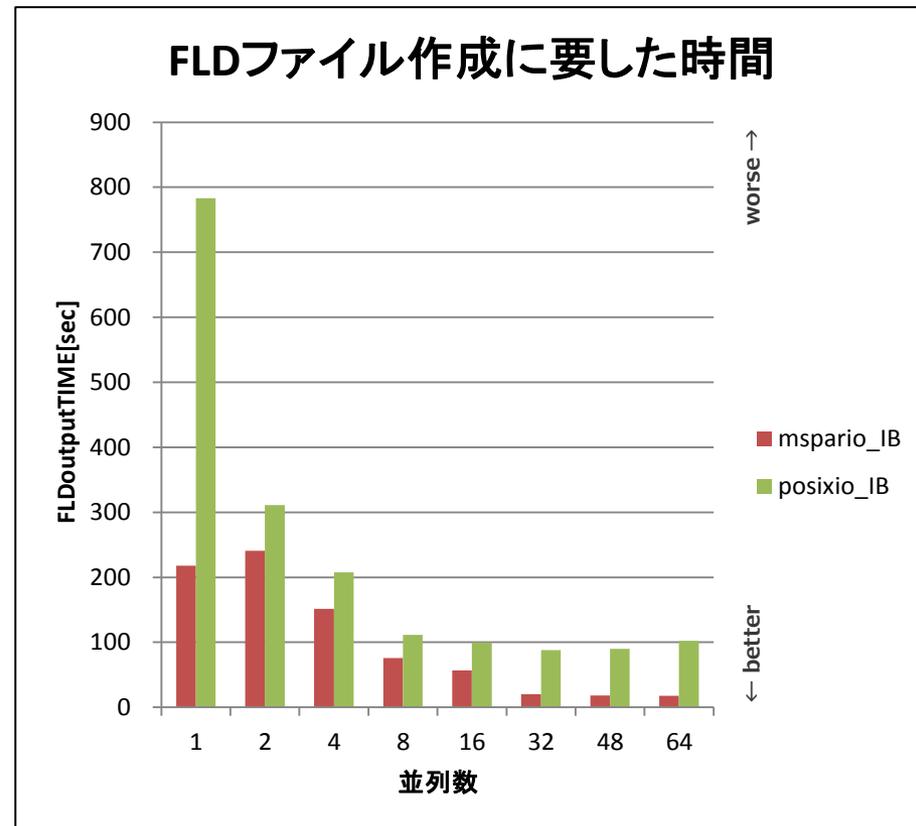
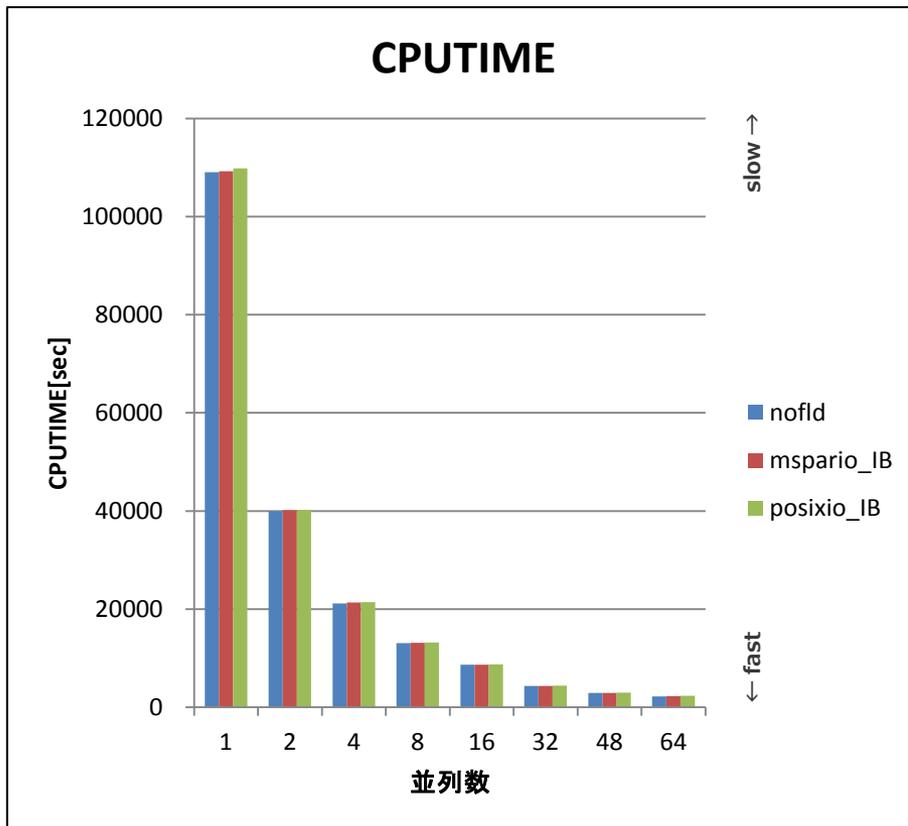
## ● Input:Propeller (10cycle)

- IB(SMBdirect)経由でFLDファイルをストレージへ書き込む
- CPUTIME = 解析開始からFLDファイル出力完了までの時間
- nofld: 同じ解析でFLDファイル出力無しの場合
- FLDoutputTIME = (各方式のCPUTIME) - (nofldのCPUTIME)  
= FLDファイルの出力に要した時間

| 並列数 | CPUTIME[sec] |            |            | FLDoutputTIME[sec] |            |
|-----|--------------|------------|------------|--------------------|------------|
|     | nofld        | mspario_IB | posixio_IB | mspario_IB         | posixio_IB |
| 1   | 109017.0     | 109235.0   | 109800.0   | 218.0              | 783.0      |
| 2   | 39953.4      | 40194.0    | 40264.3    | 240.6              | 310.9      |
| 4   | 21197.0      | 21348.4    | 21404.7    | 151.4              | 207.7      |
| 8   | 13092.1      | 13168.1    | 13203.5    | 76.0               | 111.4      |
| 16  | 8659.7       | 8716.5     | 8759.6     | 56.7               | 99.9       |
| 32  | 4338.8       | 4358.7     | 4426.6     | 19.9               | 87.8       |
| 48  | 2916.9       | 2935.0     | 3006.7     | 18.1               | 89.8       |
| 64  | 2238.3       | 2256.2     | 2340.4     | 17.9               | 102.1      |



# SCRYU/Tetra: POSIXIO vs MSPARIO

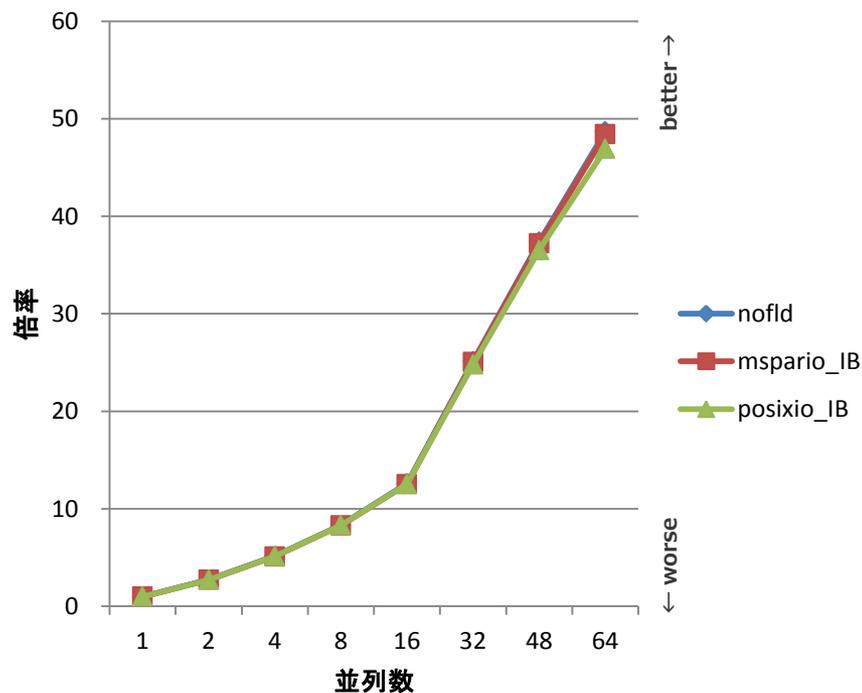


- 数値演算に費やす時間が解析の大部分を占める解析だったためCPUTIMEでは大きな差は無いが、FLDファイルの出力に要した時間はmsparioの場合がかなり少ない。

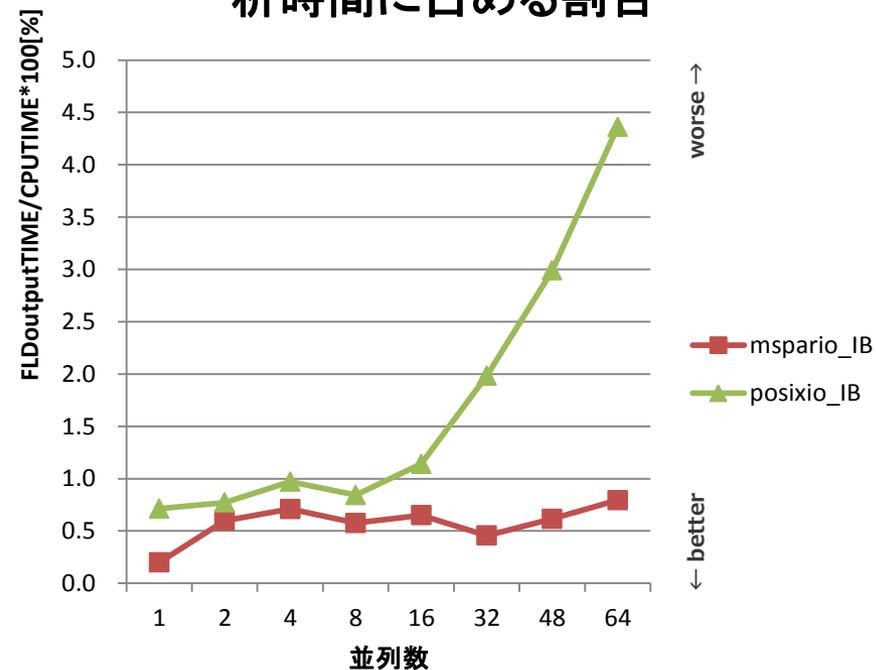


# SCRYU/Tetra: POSIXIO vs MSPARIO

スケーラビリティ



FLDファイル作成に要する時間が解析時間に占める割合



- msparioの場合並列数を増やしてもFLDファイルの出力に要する時間が解析全体に占める割合は増加しない。



# SCRYU/Tetra:1GbE vs IB(SMBdirect)

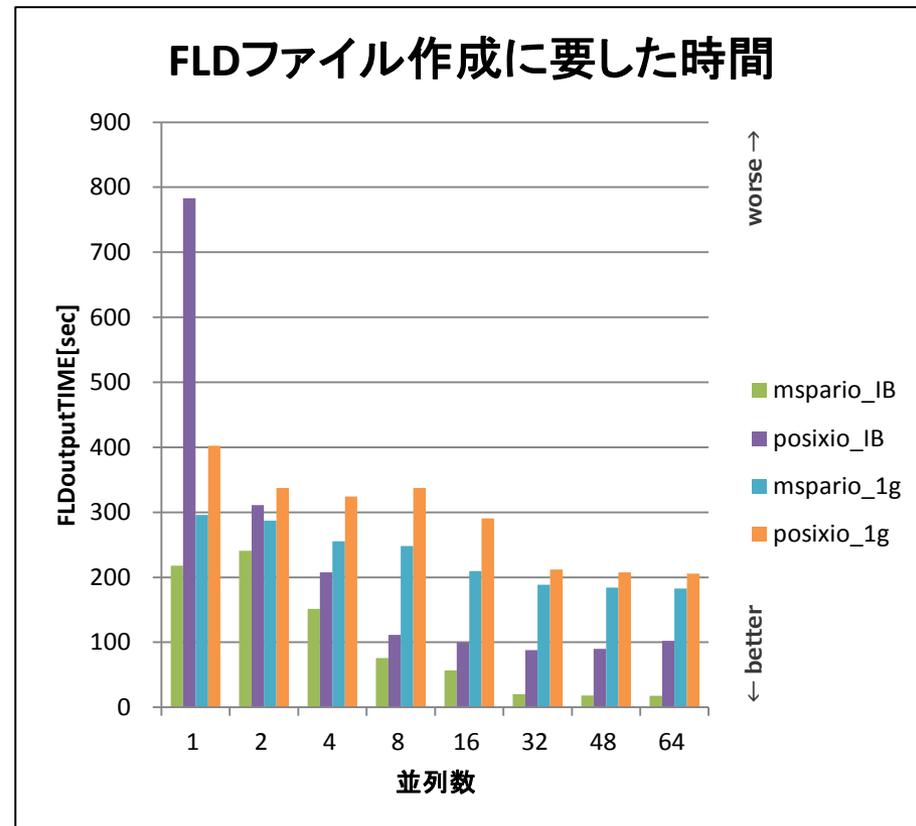
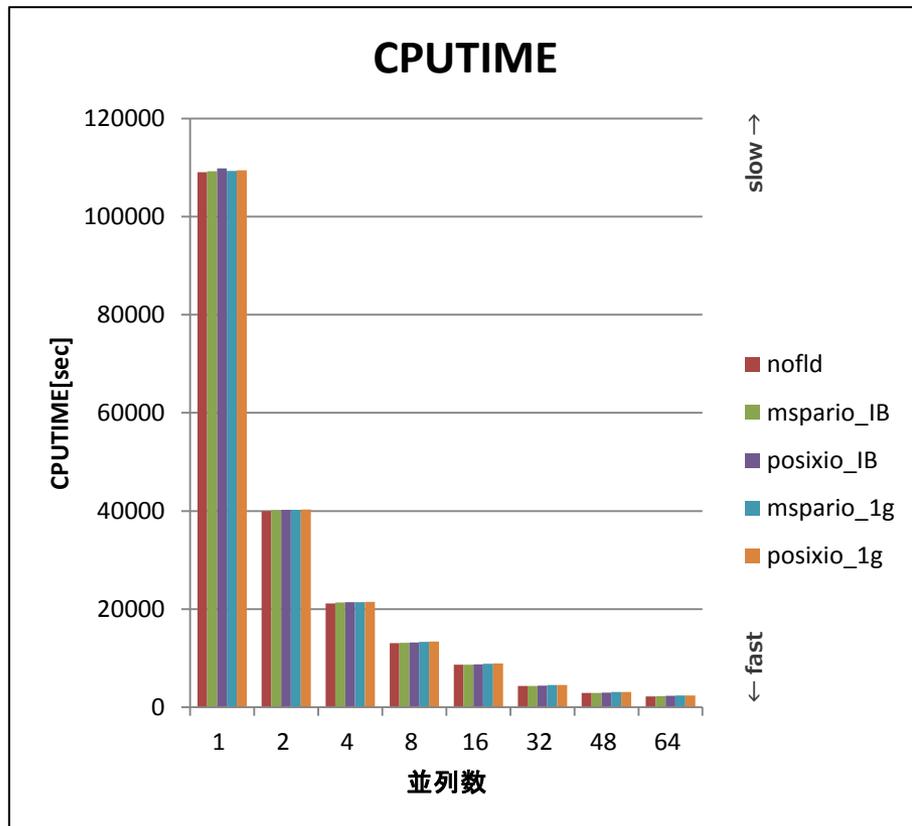
## ● Input:Propeller (10cycle)

- IB(SMBdirect)経由と1GbE経由でストレージへFLDファイルを書き込む。
- CPUTIME = 解析開始からFLDファイル出力完了までの時間
- nofld: 同じ解析でFLDファイル出力無しの場合
- FLDoutputTIME = (各方式のCPUTIME) - (nofldのCPUTIME)  
= FLDファイルの出力に要した時間

| 並列数 | nofld    | CPUTIME[sec](IB) |            | CPUTIME[sec](1GbE) |            | FLDoutputTIME[sec](IB) |            | FLDoutputTIME[sec](1GbE) |            |
|-----|----------|------------------|------------|--------------------|------------|------------------------|------------|--------------------------|------------|
|     |          | mshario_IB       | posixio_IB | mshario_1g         | posixio_1g | mshario_IB             | posixio_IB | mshario_1g               | posixio_1g |
| 1   | 109017.0 | 109235.0         | 109800.0   | 109313.0           | 109419.4   | 218.0                  | 783.0      | 296.0                    | 402.4      |
| 2   | 39953.4  | 40194.0          | 40264.3    | 40240.8            | 40290.9    | 240.6                  | 310.9      | 287.4                    | 337.5      |
| 4   | 21197.0  | 21348.4          | 21404.7    | 21452.5            | 21521.3    | 151.4                  | 207.7      | 255.5                    | 324.3      |
| 8   | 13092.1  | 13168.1          | 13203.5    | 13340.2            | 13429.4    | 76.0                   | 111.4      | 248.1                    | 337.3      |
| 16  | 8659.7   | 8716.5           | 8759.6     | 8869.4             | 8950.2     | 56.7                   | 99.9       | 209.7                    | 290.5      |
| 32  | 4338.8   | 4358.7           | 4426.6     | 4527.2             | 4550.8     | 19.9                   | 87.8       | 188.4                    | 212.0      |
| 48  | 2916.9   | 2935.0           | 3006.7     | 3100.9             | 3124.6     | 18.1                   | 89.8       | 184.1                    | 207.7      |
| 64  | 2238.3   | 2256.2           | 2340.4     | 2421.0             | 2444.1     | 17.9                   | 102.1      | 182.7                    | 205.8      |



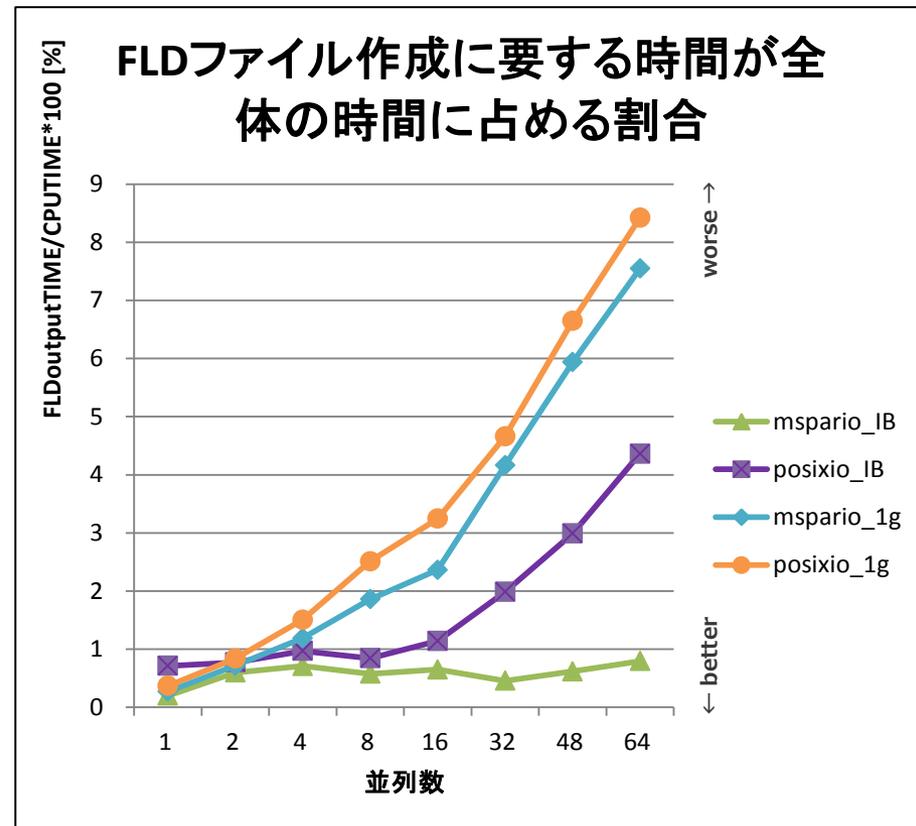
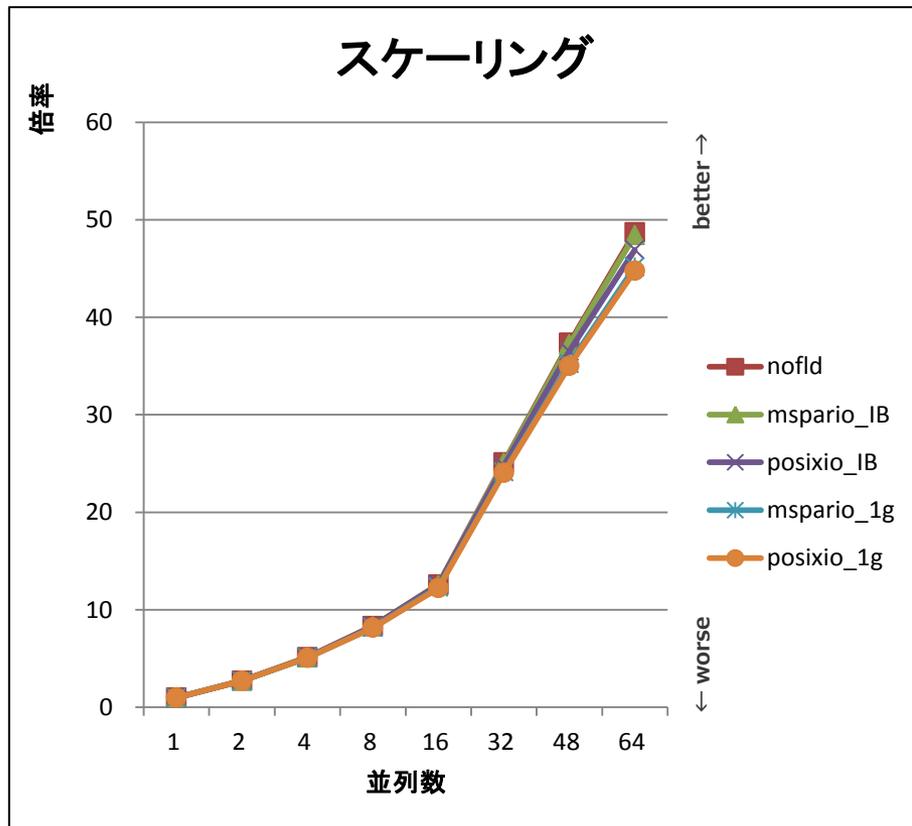
# SCRYU/Tetra:1GbE vs IB(SMBdirect)



- イタレーションが大部分を占める解析だったため、全体の時間に対しては FLDファイル作成の時間は目立たない。FLDファイルの作成に要する時間は各方式で大きな差が出ている。



# SCRYU/Tetra:1GbE vs IB(SMBdirect)

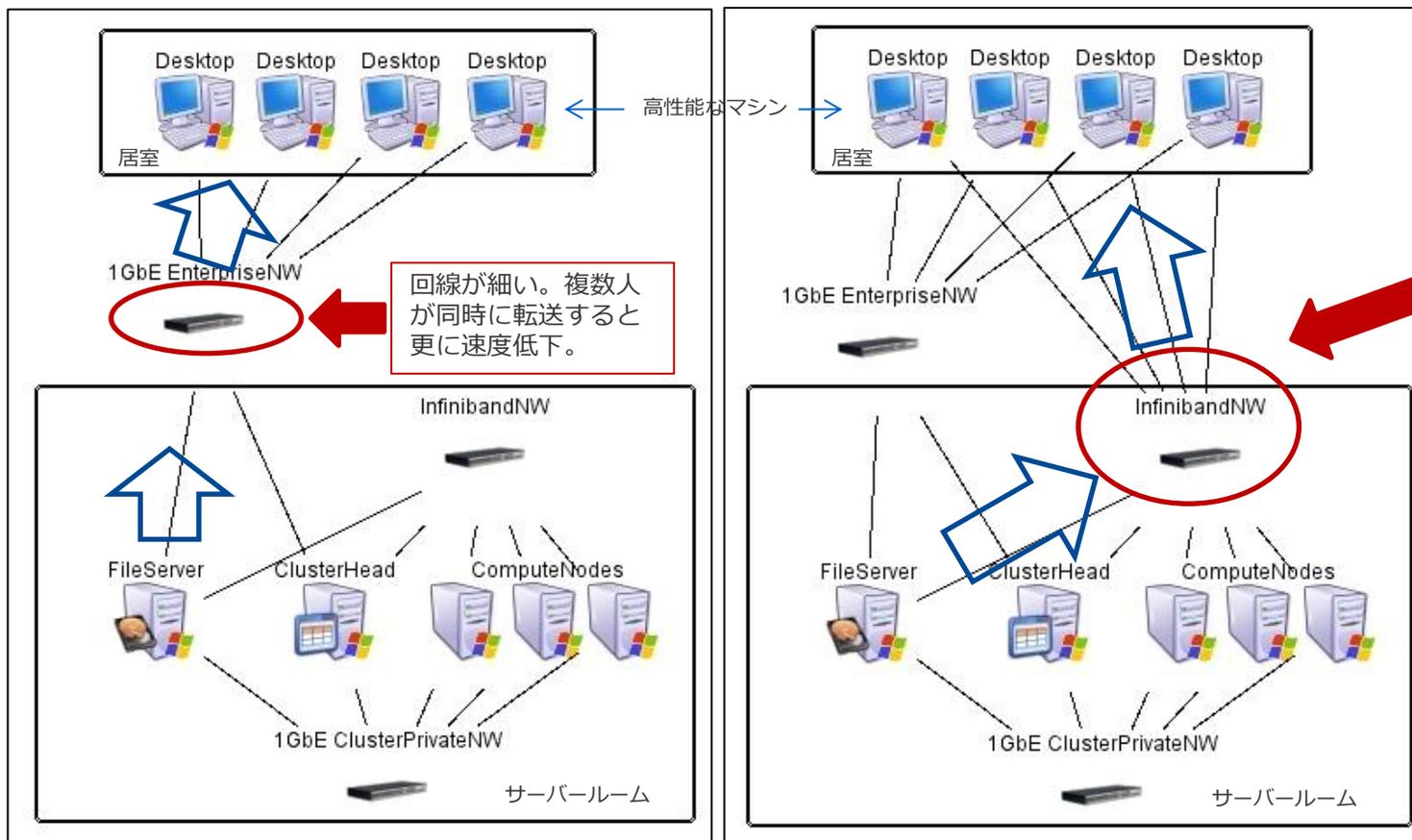


- イタレーションが大部分を占める解析だったため、全体の時間に対してはFLDファイル作成の時間は目立たない。IB-SMBdirectを用いたmsparioは並列数が増加した場合に威力を発揮できるだろう。



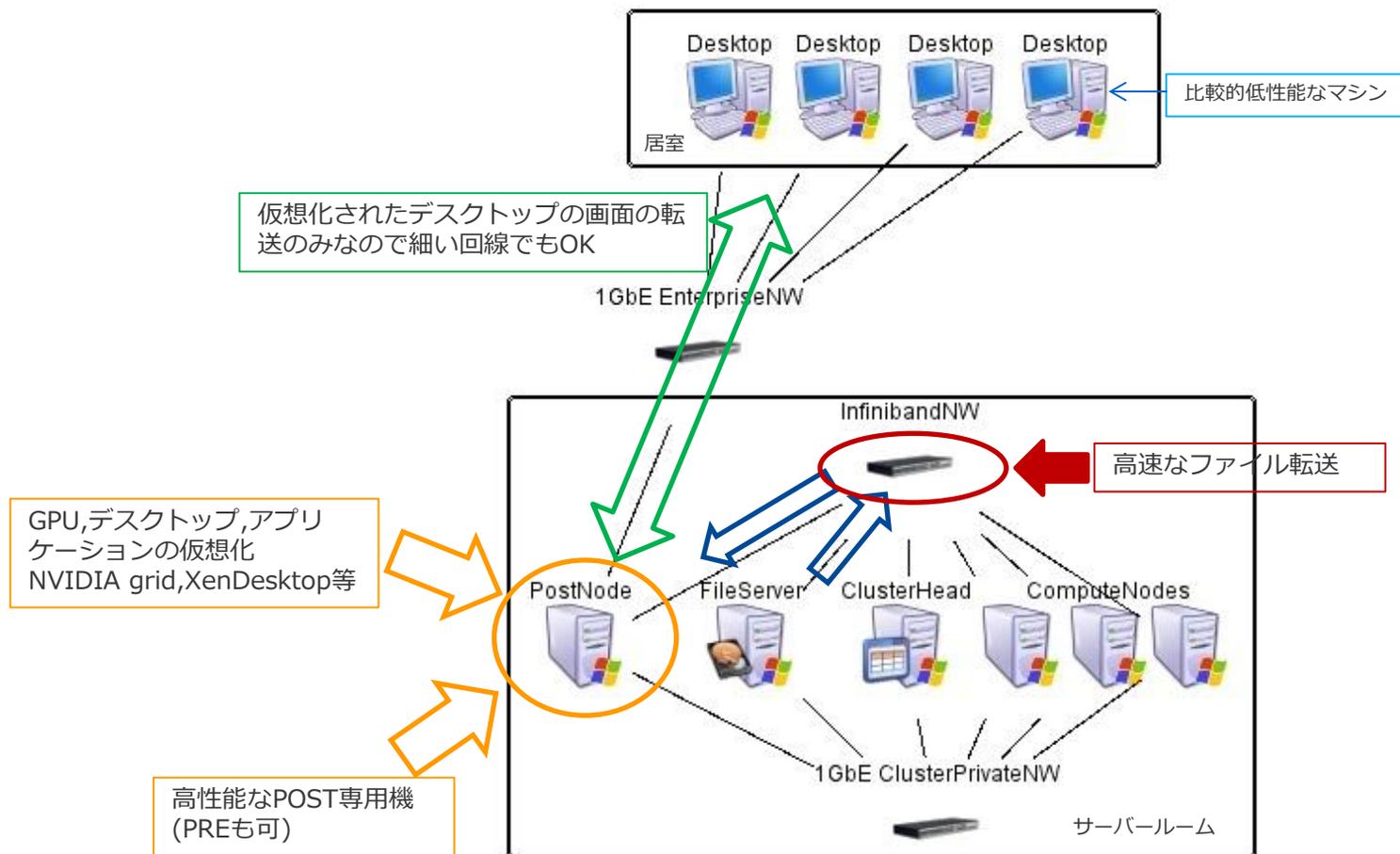
# POST処理の問題点

- POST処理のために巨大なFLDファイルを手元もPCに転送するのは時間がかかる。高速な回線は非常に高価。
- 転送してFLDファイルをPOST処理するため手元に高性能なマシンが必要。



# POST処理の問題点

## ● 改善案



# POST処理について

- **POST処理のためにファイルサーバーからPOST専用マシンへのFLDファイル読み込み時間を測定した。**

- IB経由のSMBDirectを用いた場合と1GbE経由の場合の2通りで行った。
- POSTの内部で記録している時間を用いて、総ファイルサイズ130GB程度のFLDファイルをPOSTに読み込むためにかかった時間を測定した。

1GbE : 2078.24 [sec]

IB-SMBdirect: 1138.48 [sec]

(FLDファイルの転送にかかった時間+POSTによるFLDファイルの初期処理の時間)

Infiniband経由のSMBdirectのほうが2倍弱程度速くFLDファイルを読み込み初期処理できる。また、1GbEは帯域を使い切っているため、複数のPOST処理が走った場合更に速度が低下。一方でIB-SMBdirectはまだ帯域に余裕があるため、複数のPOST処理が走った場合にも速度低下は無い。



# 課題

- SCTでFLDファイル出力にかかる時間が多いタイプの解析でベンチマークを取得したい。
- 並列数の多い環境(例えば256並列以上)でのベンチマークと安定性の確認を行いたい。
- 今回は最終的な書き込み先は超高速なSSDを用いたが、同じ環境を実現した場合高価になる。安価なHDDのRAID等を用いた場合のベンチマークも必要だと思われる。

