Vol.2017-HPC-158 No.14
2017/3/9

建屋間ネットワークのデータ転送性能評価

宇野 篤也[1,a)]　岩本 光夫[1]　八木 学[1]　横川 三津夫[2]

概要：近年，HPC システムの大規模化にともない，シミュレーション結果も膨大な量となっている．この膨大な計算結果を効率よく分析するための手段として，可視化等が用いられることが多く，可視化専用のハードウェアを搭載したシステムを利用することがよくある．この場合，シミュレーションを行ったシステムとのデータ連携が必要となる．これらのサーバが同一のサイトに設置されている場合は，ストレージ共有で対応できるが，異なるサイトに設置されているシステムを利用する場合には，ネットワーク経由でデータの転送を行うことになり，高速なデータ転送が求められる．今回，スーパーコンピュータ「京」と隣接する神戸大学統合研究拠点の計算科学教育センターに設置された可視化用計算サーバ「π-VizStudio」を直接ネットワークで接続し，データ転送性能評価を行ったので報告する．

1. はじめに

近年，コンピュータ性能の飛躍的な向上とシステムの大規模化にともない，シミュレーション結果も膨大な量となってきている．例えば，理化学研究所 計算科学研究機構（以下，AICS）が運用しているスーパーコンピュータ「京」[1] で石原らが行っている一様等方性乱流の直接数値シミュレーション (Direct Numerical Simulation; DNS)[2] では，格子点数 $12,288^3$ の場合の 1 時間ステップのシミュレーション結果は約 $41TB$ にもなる．シミュレーションで使用したシステムが混雑しており計算結果の分析用ジョブの実行が困難な場合や可視化等で専用のハードウェアを利用する場合など，シミュレーションを実施したシステムとは別のシステムを利用する場合，システム間でのデータ連携が必要となる．これらのシステムが同一のサイトに設置されている場合は，ストレージ共有でデータ連携を行うことができるが，異なるサイトに設置されているシステムを利用する場合には，ネットワーク経由でデータの転送を行うことになり，高速なデータ転送が求められる．

今回，スーパーコンピュータ「京」が設置されている AICS の敷地に隣接する神戸大学統合研究拠点の計算科学教育センターに設置された可視化用計算サーバ「π-VizStudio」[3] を「京」のシミュレーション結果の分析および可視化に利用することを目的に SINET5 を経由せず直接ネットワークで接続しデータ転送性能評価を行ったので報告する．

1　国立研究開発法人理化学研究所 計算科学研究機構
2　神戸大学
a)　uno@riken.jp

2. 「京」と「π-VizStudio」

「京」は，82,944 台の計算ノードと 1.27PiB のメモリ，11PB のローカルファイルシステム (LFS) と 30PB 超のグローバルファイルシステム (GFS) 等から構成され，SINET5 へは 100GB/s で接続している（**図 1**）．ユーザのプログラムやデータを保存するストレージには，富士通が Lustre をベースに機能拡張を行った FEFS (Fujitsu Exabyte File System) が採用されている．

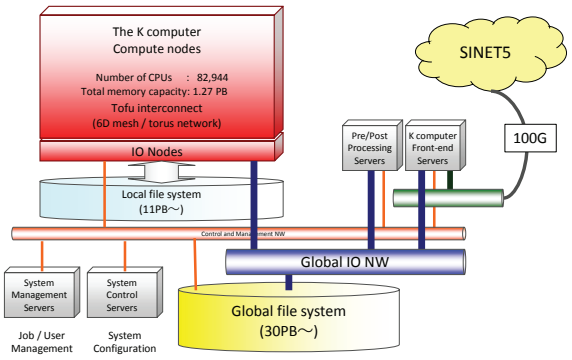

図 1　「京」の構成

「π-VizStudio」は，スーパーコンピュータ「京」が設置されている AICS の敷地に隣接する神戸大学統合研究拠点の計算科学教育センターに設置されたテラスケールのシミュレーションデータのポスト処理・解析が可能なシステムである（**図 2**，**図 3**）．同センター内に設置されている 3 次元可視化システム π-CAVE と連携し，可視化したシミュレーションデータを見ながらマルチモーダル・インターフェー

スを経由して，可視化パラメータの変更や時間の操作などを対話的に行うことができる．ユーザのデータを保存するファイルシステムは xfs で，ログインノードの vizfront からは nfs でマウントされている．

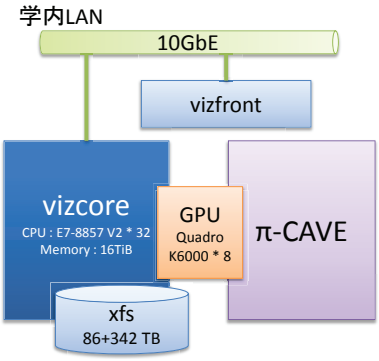

図 2 「π-VizStudio」の構成

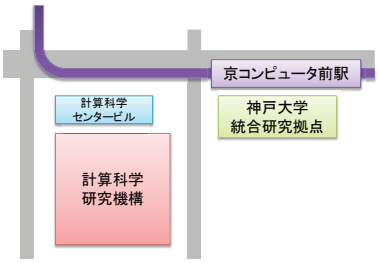

図 3 AICS と神戸大学統合研究拠点

「π-VizStudio」は神戸大学の学内 LAN 内に設置されているため，SINET5 へは学内 LAN を経由してアクセスする．そのため，「京」からのアクセスは学内 LAN を経由することになり，データ転送に十分な帯域を確保することが難しい．そこで，学内 LAN を経由せず「京」と「π-VizStudio」を直接接続することとした．**図 4** にネットワーク構成を示す．「京」と「π-VizStudio」の各ルータ間を 10GbE で直接接続し，「京」と「π-VizStudio」間の通信のみこの 10GbE を経由するようにルーティングを設定した．

今回，この 10GbE を経由した「京」と「π-VizStudio」間の通信について通信性能を測定した．

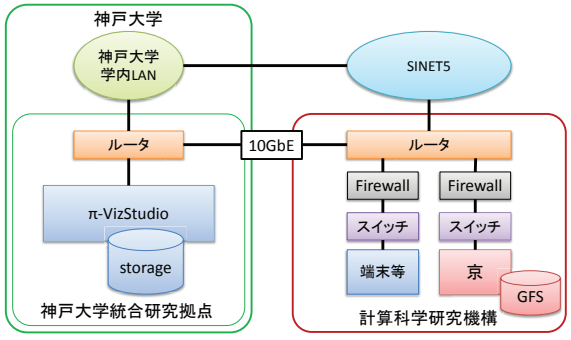

図 4 ネットワーク構成

3. ネットワーク設定

図 5 に「京」のフロントエンドサーバ klogin と「π-VizStudio」の vizfront 間の接続構成を示す．それぞれのルータ間を直接接続した．

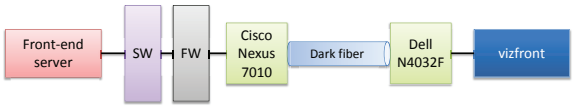

図 5 「京」と「π-VizStudio」間の接続構成

「京」は既に運用を行っていたため，「京」側のネットワークに影響がでないようにネットワークの設定を行う必要があった．そのため，サーバ及びクライアント側で設定可能なパラメータについてのみ調整を行うこととした．今回の性能測定で調査したパラメータを**表 1** に示す．MTU(Maximum Transmission Unit) はデフォルト値の 1500B とした．MTU は一般的に大きくすることでパケット化のオーバーヘッドを軽減し転送性能を改善できるが，MTU の変更は多数の既存機器のインターフェースの設定変更が必要なことと，最近の機器の高性能化によりデフォルト値でも十分に性能がでることがわかっていることから値の変更は見送った．

TCP 通信バッファサイズを決めるにあたり，帯域幅遅延積 (Bandwidth Delay Product:BDP) を求めた．BDP は，リンク帯域幅と往復遅延時間 (Round-Trip Time:RTT) から以下の式で求められる．

$BDP =$ リンク帯域幅 $\times RTT$

klogin と vizfront 間の RTT を測定したところ平均で $0.246ms$ であった．BDP は

$BDP = 10Gbps \times RTT/8 = 330,175B$

となるので，TCP 通信バッファとして 約 330KiB あれば理論性能を出せることになる．

転送性能の測定に使用した klogin と vizfront の仕様を**表 2** に示す．転送性能は scp によるファイルコピーで測定した．vizfront の ssh は OpenSSH 5.3p1 がインストールされているが，scp は klogin と同じ OpenSSH 7.2p2 を独自にコンパイルし使用した．各ファイルのサイズは 1GiB で，並列数を変えて転送性能を測定した．ストレージの read/write 性能の影響を減らすため，転送元のファイルは tmpfs 上に作成し，転送先は /dev/null に書き込むこととした．

4. 測定結果

図 6 にバッファサイズと並列数を変えて測定した結果を示す．暗号スイート [*1] は aes128-ctr で，並列数は 1,4,8,12 並列，バッファサイズは 16KiB, 32KiB, 64KiB, 128KiB,

[*1] 認証・鍵交換・暗号化・MAC のアルゴリズムの一式

表 1 チューニングパラメータ

項目	説明
net.ipv4.tcp_mem	TCP がメモリ使用量を追跡する際に使用するサイズ
net.ipv4.tcp_wmem	TCP の送信バッファのサイズ
net.ipv4.tcp_rmem	TCP の受信バッファのサイズ
net.core.wmem_max	TCP の送信バッファの最大サイズ
net.core.rmem_max	TCP の送信バッファの最大サイズ
net.core.wmem_default	TCP の送信バッファのデフォルトサイズ
net.core.rmem_default	TCP の受信バッファのデフォルトサイズ
net.core.optmem_max	補助バッファの最大サイズ

表 2 測定に用いたサーバの仕様

	「京」		「π-VizStudio」
名称	klogin	klogin2	vizfront
OS	RHEL 6.5	RHEL 6.5	RHEL 6.8
CPU	Intel Xeon E5620 × 2 (4core/2.4GHz/12MiB)	Intel Xeon E5-2697v3 × 2 (14core/2.60GHz/35MiB)	Intel Xeon E5-2667v3 × 2 (8core/3.2GHz/20MiB)
メモリ	72GiB	128GiB	256GiB
ネットワーク	10Gbps	10Gbps	10Gbps
ssh	OpenSSH 7.2p2 OpenSSL 1.0.1e-fips	OpenSSH 7.2p2 OpenSSL 1.0.1e-fips	OpenSSH 5.3p1 (sshd) OpenSSH 7.2p2 (scp) OpenSSL 1.0.1e-fips

192KiB, 256KiB, 384KiB, 512KiB である．図中 K klogin → vizfront は，klogin 上で scp を実行し klogin から vizfront へファイルを転送したことを，V vizfront → klogin は，vizfront 上で scp を実行し vizfront から klogin へファイルを転送したことをそれぞれ示している．K klogin → vizfront の 1 並列の転送性能のピークは約 220MiB/s で，バッファサイズが 330KiB よりも小さい段階で性能が低下していることがわかる．330KiB は CPU 性能を考慮せず帯域と RTT から求めた値であり，330KiB よりも小さいバッファサイズで性能が低下したということは，CPU 性能がボトルネックになっていることを示していると考えられる．理論性能が 220MiB/s の時のバッファサイズは

$$220MiB/s \times 0.246ms = 56,748B$$

であり，これはバッファサイズが 64KiB でピーク性能となっている結果と一致する．並列数については，12 並列で 1200MiB/s 近くでており，理論性能に近い値が得られていることがわかる．

一方，klogin 上での転送と vizfront 上での転送を比較すると，転送方向の傾向は類似しているが転送性能では 2 倍近い差がでている．また，scp を実行するサーバに関係なくバッファサイズが小さい時の vizfront → klogin 方向の転送性能が低い．klogin よりも vizfront の方が CPU 性能が高く，vizfront → klogin と klogin → vizfront の RTT もほぼ同じでネットワークには問題はみあたらなかったため，scp に何か問題がないか調査を行うことにした．

まず OpenSSH の暗号化性能を測定した．暗号スイートの arcfour，aes128-ctr，aes256-ctr について，ファイル (1GiB) の転送性能を測定した．バッファサイズは影響を受けない十分に大きい値とし，CPU 性能の影響を調べるため，klogin よりも CPU 性能の良いサーバ (以下，klogin2．Intel Xeon E5-2697v3(14core/2.60GHz/35MB)×2，128GiB) 間での測定も行った．**図 7** に結果を示す．klogin/klogin2 では，arcfour は受け付けない設定になっているため，vizfront 上での測定はできていない．

arcfour を使用した場合は klogin2 の方が klogin より良い結果となっており，klogin では CPU 性能がボトルネックになっていることがわかる．一方，aes の場合は図 6 と同じように，login/klogin2 で scp を実行した場合に vizfront 上で実行した場合に比べて 2 倍近い性能がでている．また，klogin/klogin2 上で scp を実行した場合の転送性能は aes128 ＞ aes256 でベンチマークと同じ傾向になっているが，visfront 上で scp を実行した場合は aes128　aes256 となっており，vizfront の scp か klogin/klogin2 の sshd の暗号化処理がボトルネックになっていると考えられる．そこで，openssl のベンチマーク (openssl speed aes rc4 -multi 1) で暗号化の性能を測定した．測定結果を**図 8** に示す．cbc(Cipher Block Chainig) と ctr(CounTeR) を単純に比較することはできないが，この測定では CPU 性能に応じた結果が得られた．

次に sshd の性能を調べるために，自分自身への転送性能を測定した．**図 9** に自分自身へ 1GiB のファイルを転送した場合の測定結果を示す．put は転送データがローカルにある場合を，get は転送データをローカルにコピーする場合を示している．図 9 と図 7 を比較すると，klogin/klogin2

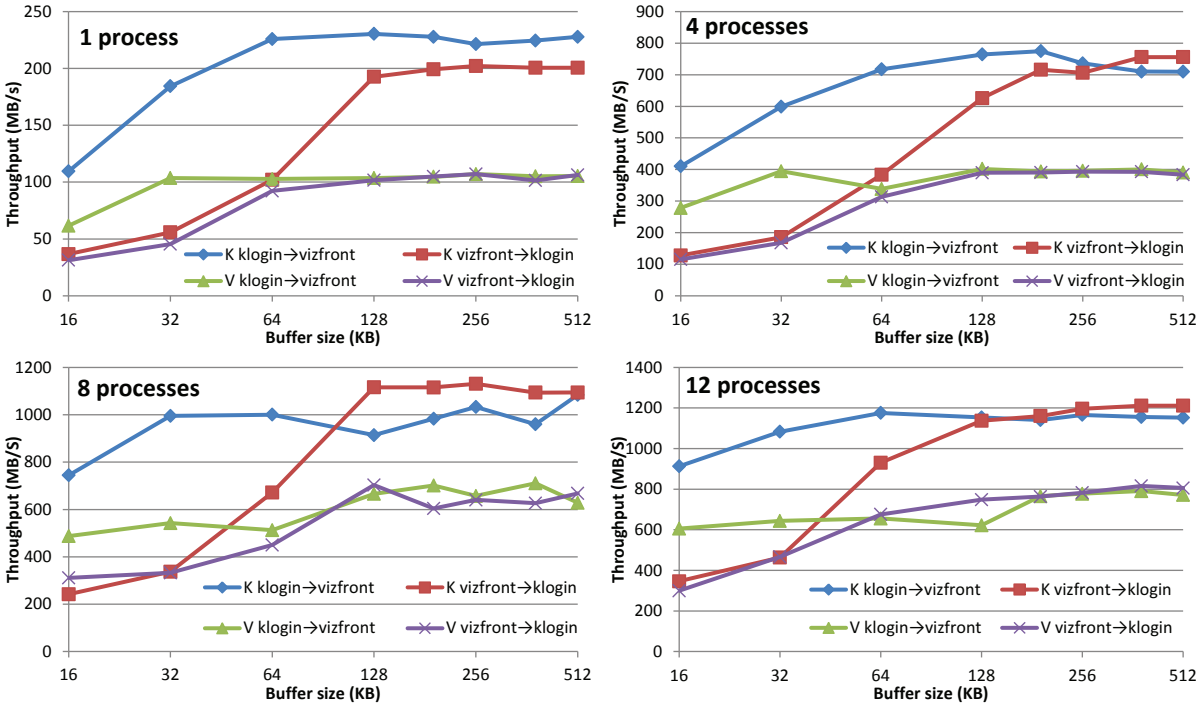

図 6 転送性能の測定結果

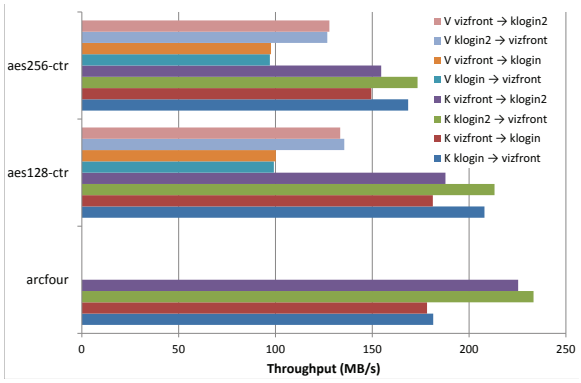

図 7 暗号スイート別の測定結果

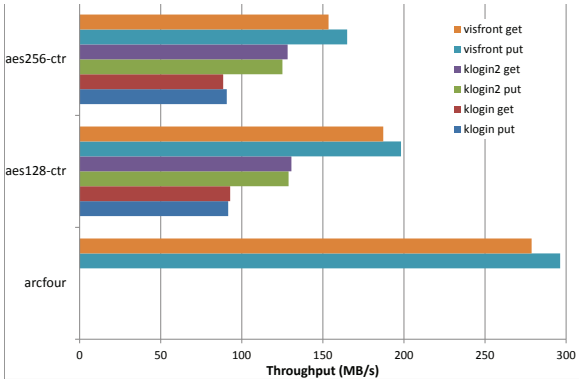

図 9 自分自身へ転送した場合の測定結果

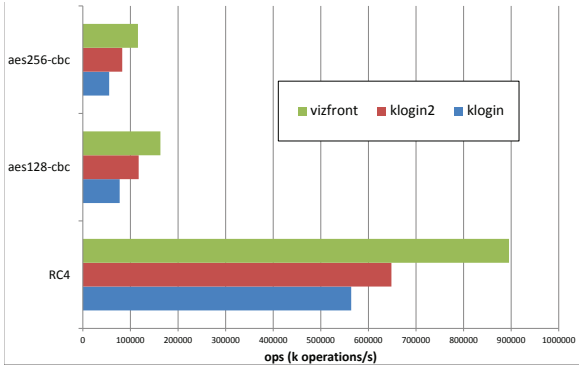

図 8 openssl のベンチマーク結果

の結果が vizfront 上で scp を実行した際の結果と一致していることがわかる．このことから，klogin/klogin2 の sshd の性能に問題があると推測される．この性能低下の原因が CPU 性能によるものなのか，他にあるのか現在調査中である．

5. おわりに

今回，スーパーコンピュータ「京」と隣接する神戸大学計算科学教育センターに設置された可視化用計算サーバ「π-VizStudio」を直接ネットワークで接続し，データ転送性能評価を行った．システム間はネットワーク的に近距離にあるためサーバの CPU 性能がボトルネックとなったが，並列度を上げることで理論値に近い転送性能を出すことが可能であった．ただ，「π-VizStudio」から「京」へのデータ転送では十分な性能を出すことができておらず，今後の課題である．

参考文献

[1] 特集：スーパーコンピュータ「京」, 情報処理, Vol.53, No.8, pp.752–807, 2012.

[2] T. Ishihara, K. Morishita, M. Yokokawa, A. Uno, and Y. Kaneda: Energy spectrum in high-resolution direct numerical simulations of turbulence, Physical Review Fluids, Vol.1, No.8, 2016.
[3] http://www.eccse.kobe-u.ac.jp/pi-vizstudio/