

# VirtualGL を用いたリモート可視化環境の改善

疋田 淳一<sup>1)</sup>, 當山 達也<sup>1)</sup>, 尾形 幸亮<sup>1)</sup>

1) 京都大学 企画・情報部

junichi.hikita.5n@kyoto-u.ac.jp

## Improvement of remote visualization environment with VirtualGL

Junichi Hikita<sup>1)</sup>, Tatsuya Tohyama<sup>1)</sup>, Kosuke Ogata<sup>1)</sup>

1) Planning and Information Management Department, Kyoto Univ.

### 概要

京都大学学術情報メディアセンターでは、VirtualGL のサーバサイドレンダリングにより、スーパーコンピュータシステムにおけるリモート可視化環境を改善することを検討している。本稿では、VirtualGL を用いた場合の動作検証および性能の調査結果について紹介する。

## 1. はじめに

京都大学学術情報メディアセンター（以下、本センターという）では、全国の研究者に対しスーパーコンピュータシステムと共に、可視化やポスト処理のためのソフトウェアを提供している。Linux の GUI をパソコンに転送するには、X Window System が必要となるため、本センターでは OpenText 社の Exceed onDemand[1]を利用者に提供することで対応している。この他に、SSH の X11 Forwarding 機能により画面を転送することも可能であり、パソコン側に X Window System の 1 つの実装である Windows 用の Xming[2]や macOS 用の XQuartz[3]等を別途導入することで GUI を利用できる。さらに別の方法として、VNC によるリモートデスクトップ接続も可能である。

いずれの方法であっても、リモート可視化を行う場合、3D データの描画能力が重要となるが、パソコン側の処理能力への依存やソフトウェアレンダリングにより処理されることで十分な応答性を得られない状況が発生するケースがある。そこで、本センターでは、VirtualGL[4]を用いて、サーバに搭載した GPU により 3D データのレンダリングを行い、クライアントには画像として転送するサーバサイドレンダリングを実現し、クライアントの性能に極力依存しないリモート可視化環境を提供するための改善を検討している。本稿では、VirtualGL を用いたサーバサイドレンダリング及び X Window System や VNC との組み合わせについて調査した結果について紹介する。

## 2. VirtualGL の概要

VirtualGL は、サーバ上で 3D データの描画を行い、クライアントに画像データを送出する仕組みを実装したオープンソースソフトウェアである。VirtualGL のライブラリを可視化ソフトウェアの起動時にプリロードすることで、3D の描画命令への介入が行われ、強制的な GPU のハードウェアアクセラレーションが実現される。3D の描画に関してのみ介入し、2 次元データの描画には影響を与えない方式であるため、X11 Forwarding や VNC との親和性が高く、既存の方法に影響を与えずに実現可能である点が優れている。

利用するためには、サーバに VirtualGL のインストールが必要となるが、Linux 向けのインストールパッケージが用意されており、主要な Linux ディストリビューションであれば容易にインストール可能である。vglserver\_config コマンドにより、サーバ上の設定を行った後に、vglrun コマンドを介して可視化ソフトウェアを起動することでサーバサイドレンダリングが可能となる。VirtualGL に対応しているソフトウェアであれば、オプション等により、vglrun コマンドの実行がオプションとして実装されているケースもある。

### 3. VirtualGL とリモート接続環境

リモートの Linux サーバの GUI アプリケーションを操作するためには、SSH の X11 Forwarding または VNC によるリモートデスクトップを使う方法が一般的である。X11 Forwarding の場合はローカル環境に X Window System が必要であり、VNC の場合は VNC サーバ及びクライアントが必要となる。それぞれオープンソースや有償のソフトウェアが複数存在しているため、本章で述べるソフトウェアにおいて VirtualGL によるサーバサイドレンダリングが可能であるかを確認した。

#### 3.1. Exceed onDemand

OpenText 社の Exceed onDemand は、専用のクライアントで接続することで X11 環境を利用可能な有償ソフトウェアであり、Windows、Mac、Linux に対応している。フローティングライセンスであれば、クライアントのインストール台数では制限されず、同時利用数により制限されるため、本センターのような多人数でシステムを共有利用する形態でも対応可能である。本センターでは Exceed onDemand をすでに導入しているため、VirtualGL と併用することで、GPU 搭載サーバさえ用意すれば簡易にサーバサイドレンダリングの実現が可能である。Exceed onDemand の認証方式はパスワード認証のみに対応しているため、セキュリティの強度を考えると SSH ポートフォワードとの併用が必要となる。

#### 3.2. FastX

StarNet 社の FastX[5]は Exceed onDemand と同様な機能およびライセンス形態を持つソフトウェアであるが、SSH の公開鍵認証にも対応している点、WEB ブラウザをクライアントとした接続に対応している点が優れているソフトウェアである。なお、本稿においては、WEB ブラウザによる接続の確認は省略している。

#### 3.3. Xming および XQuartz

オープンソースの X Window System の実装として、Windows 向けに Xming、macOS 向けに XQuartz がある。どちらも各 OS において利用者

の多い X11 環境であると考えられるため、調査対象に加えている。

#### 3.4. TurboVNC

TurboVNC[6]は、3D データの描画性能を強化する目的で TigerVNC から派生したオープンソースの VNC の実装である。クライアントソフトは Windows、Mac、Linux に対応している。VirtualGL との併用を念頭に置かれて開発されており、VirtualGL を有効にして VNC サーバを起動するオプションが用意されている等、VirtualGL を VNC で利用する場合の第一候補であると考えられる。他の VNC 実装と同様に VNC サーバへの接続には専用のポートにパスワード認証で接続する必要があるため、安全を考えると SSH の公開鍵認証およびポートフォワードと併用することが必要となる。

#### 3.5. 動作検証結果

VirtualGL の配布パッケージにデモプログラムとして含まれている `glxspheres64` コマンドを、前述した 5 種のリモート接続環境において起動させることで VirtualGL の動作検証を行った。動作検証結果を表 1 に示す。起動時には `vglrun` コマンドを用いて VirtualGL を介した起動を行った結果、いずれの環境においても不具合は生じなかった。

表 1 動作検証結果

	Windows	macOS
Exceed onDemand8J 13.8.6	○	○
FastX 2-2.4-13	○	○
Xming 6.9.0.31	○	-
XQuartz 2.7.11	-	○
TurboVNC 2.1.2	○	○

### 4. 性能測定

次に VirtualGL を使うことで得られる 3D データの描画性能を測定し、十分な性能を得られるか確認した。

#### 4.1. 性能測定環境

性能測定のクライアントには Windows のノートパソコンおよび MacBook Pro を使用した。構成を表 2 及び表 3 に示す。可視化用のサーバに

は、NVIDIA 社の QuadroP4000 を搭載しているサーバを使用した。構成を表 4 に示す。

表 2 Windows パソコンの構成

OS	Windows 10
CPU	Intel Core i7-6500U 2.5GHz
GPU	Intel HD Graphics 520
メモリ	16GB

表 3 MacBook Pro の構成

OS	macOS 10.13
CPU	Intel Core i5-3210M 2.5GHz
GPU	Intel HD Graphics 4000
メモリ	16GB

表 4 サーバの構成

OS	RedHat Enterprise Linux 7
CPU	Intel Xeon Gold 6140 2.30GHz
GPU	NVIDIA Quadro P4000
メモリ	256GB

#### 4.2. Unigine Valley ベンチマーク

3D データの描画性能を測るベンチマークとして Unigine 社の Valley ベンチマークを使用した。OpenGL を用いて実装されており、GPU に高い負荷がかかるベンチマークであるため、サーバで直接実行した場合とリモート実行における性能差が見えやすいと考えている。Valley ベンチマークの条件設定として、1280x720 の Windows サイズで描画する Basic 設定を選択した。

#### 4.3. 測定結果

リモート接続方法およびパソコンの OS 毎の実行結果を図 1 に示す。性能値として、ベンチマーク結果に表示される平均 FPS を使用した。サーバのローカル環境で直接実行した際の FPS が 103 なのに対し、TurboVNC+Windows の環境においては、56FPS という約半分の性能が得られている。Exceed onDemand の 53FPS、FastX の 40FPS という値についても、リモート可視化の性能としては実用に耐える値であると考えられる。Xming は 4FPS という低い結果となったが、VirtualGL が高速化に使う X11 の拡張機能に未対応であることが原因と考えられる。macOS の環境については、

Windows パソコンと構成が異なるので直接比較はできないが、リモート可視化としては支障のないレベルの応答性が得られていると考えられる。なお、macOS 環境における Exceed onDemand の結果は、ベンチマークソフトが途中でフリーズしたため結果が得られていない。

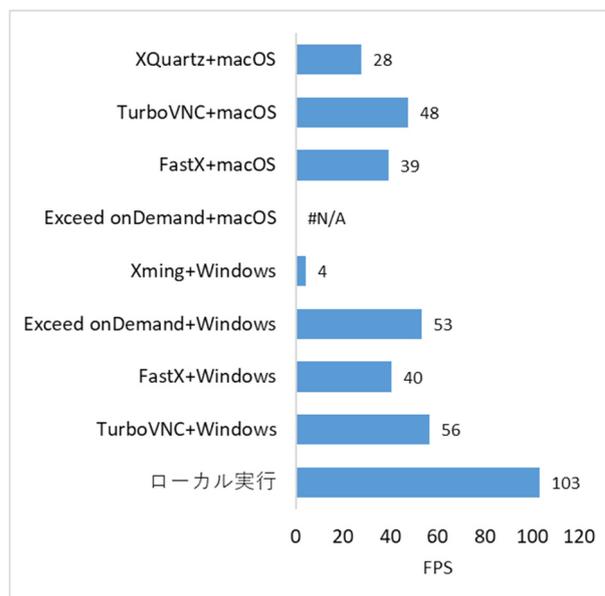


図 1 ベンチマーク結果

### 5. おわりに

今回の調査により、VirtualGL を用いることでサーバに搭載した GPU によるリモート可視化が実現可能であることを確認した。今後、評価に用いたサーバをスパコン利用者に提供することで可視化環境の改善を進める予定である。

### 参考文献

- [1] OpenText Exceed onDemand  
<https://www.macnica.net/opentext/eod.html/>  
(2018 年 8 月 8 日閲覧)
- [2] Xming  
<http://www.straightrunning.com/XmingNotes/>  
(2018 年 8 月 8 日閲覧)
- [3] XQuartz  
<https://www.xquartz.org/>  
(2018 年 8 月 8 日閲覧)
- [4] VirtualGL  
<https://www.virtualgl.org/>  
(2018 年 8 月 8 日閲覧)
- [5] StarNet FastX  
<https://www.starnet.com/fastx/>  
(2018 年 8 月 8 日閲覧)
- [6] TurboVNC  
<https://www.turbovnc.org/>  
(2018 年 8 月 8 日閲覧)