

## グリッドコンピューティング環境における 遠隔可視化処理の高速化に関する研究

田中拓也\* 江原康生† 曾根秀昭\*\* 小山田耕二‡

\* 東北大学大学院情報科学研究科 † 京都大学学術情報メディアセンター

\*\* 東北大学情報シナジーセンター ‡ 京都大学高等教育研究開発推進センター

科学技術計算による大規模データの遠隔可視化に対する要求の一つとして、高速化が挙げられる。しかし、高速化を図る際のボトルネックの要因として、ネットワークにおける伝送遅延が考えられる。本研究では遠隔可視化の高速化を目的として、データ転送時間の削減を考える。具体的には、TCP を用いた場合のネットワーク転送における問題点を明確にし、その解決策の一案として、高速性と信頼性を有するプロトコルである RBUDP(Reliable Blast UDP) を我々の構築する遠隔可視化システムに実装した。さらに、東北大学と京都大学間を Super SINET で介した長距離ネットワーク環境で能評価実験を行った。本稿では、その概要について述べる。

## Study on Speed-up for Remote Visualization in Grid Computing Environment

Takuya TANAKA\* Yasuo EBARA† Hideaki SONE\*\* Koji KOYAMADA‡

\* Graduate School of Information Sciences, Tohoku University

† Academic Center for Computing and Media Studies, Kyoto University

\*\* Information Synergy Center, Tohoku University

‡ Center for the Promotion of Excellence in Higher Education, Kyoto University

High-speed performance is one of quality which requested for remote visualization with large-scale science and technology computing. However, it is considered that transmission delay of network is a bottleneck factor. In this research, we discuss the reduction of data transmission in order to realize high-speed remote visualization. Concretely, we make clear some problems of visualization data by using TCP protocol, and as one of the solutions, we implemented RBUDP(Reliable Blast UDP) protocol which is high-speed quality and reliance quality with our remote visualization system. Moreover, we have evaluated in long-distance network environment that Tohoku University and Kyoto University are connected to Super SINET. In this report, we have expressed these contents.

## 1 はじめに

近年、計算機の処理能力の向上に伴い、高度かつ大規模なシミュレーションが可能となり、その結果をわかりやすく見せる手段として、情報可視化に対する要望が増えている。しかし、大規模な可視化システムは限られた研究機関だけしか設置されておらず、また必ずしもユーザの満足する処理方法を達成できる性能を有していない場合も少なくない。そのため、遠隔地に点在する可視化システムのリソースを有効活用できるような遠隔可視化環境の整備が必要とされる。さらに高速ネットワーク網の整備が進む中、遠隔地間で可視化データを共有しながら研究者間で共通の問題に取り組む遠隔協調作業が非常に有益な方法である。しかし現状は、大規模な可視化データの遠隔地間での実時間転送及び共有技術は十分ではなく、これらの実現に向けた効率的な大規模データの遠隔可視化技術の開発が重要と考える。

このような要求に対して、我々は処理の一部を可視化対象データを有する遠隔地の並列型計算サーバで行い、クライアント側に市販のハードウェアグラフィックエンジンと2次元画像重畳装置を使った可視化クラスタを構築し、レンダリング処理を表示装置の存在する地点で行うシステムによって遠隔可視化を実現している [1]。並列計算サーバに処理の一部を分散させることによって、可視化対象データの共有や処理の高速化が期待できる。また、クラスタを構成してレンダリング処理を分割して行うことで各ノードにおけるスワッピングを回避することができ、クライアント側でも処理の高速化を図ることができる。

一般的な遠隔可視化システムは、クライアント/サーバモデルで実現され、可視化を行う際、サーバにおける処理、データ転送、クライアントにおける処理の3つのフェーズが遂行される。しかし遠隔可視化の高速化を目指す際、最大の障害となっているのは長距離ネットワークにおけるデータの転送時間である。遠隔間で通信を行う際は、距離による遅延の影響が大きくなる。さらに、高速なネットワークを用いる場合、単位時間あたりの無駄が大きくなり、その影響は無視できないものとなり、これはLFN(Long Fat Networks)における問題点として知られている。

我々が構築した従来の遠隔可視化システム [1][2] では、トランスポートプロトコルとしてTCPとUDPを実装していたが、UDPは信頼性が低く、パケットの損失が発生するとデータの損失を補うことができ

ないため、クライアント側にデータが正確に転送されずにレンダリング処理が行えない深刻な問題があり、主にTCPを使用していた。しかしTCPでは、ACKが帰ってくるまでの間はデータを転送することができないので、遅延の影響を受けるため、可視化の高速性を追求するには満足できないと考える。

本稿では遠隔可視化処理の高速化を目的とし、高速性と信頼性を有するトランスポートプロトコルであるRBUDP(Reliable Blast UDP)[3]を我々の遠隔可視化システムに実装した。そして、東北大学と京都大学間の長距離ネットワーク環境で性能評価実験を行った。その概要について報告する。

## 2 RBUDP

TCPは信頼性を有するプロトコルであり、ACKを介してパケットの到達有無を確認し、パケットロスが生じると再送制御を行う。確認応答はウィンドウサイズごとに行われ、ウィンドウ制御によって通信速度の向上が図られている。しかし、確認応答に費やす時間は長距離ネットワークにおける伝送では大きな遅延が生じる。さらにネットワークを混雑させないための輻輳制御も行われているため、リアルタイム性が求められる通信には適さない。

一方、UDPは確認応答などを行わずに通信するため、TCPよりも高速なデータ転送が期待できる。しかし、UDPはパケットが失われても再送制御は行われない。そのため、可視化処理のように転送されるデータの欠落が生じることで、画像表示側でのレンダリング処理が実行できなくなる問題を引き起こす。つまり遠隔可視化といったデータ転送の信頼性を必要とする通信ではUDPの利用には適さないと考える。

本システムでは、長距離間の大容量データ転送を高速に、かつデータ損失を起こさずに転送するために、高速性と信頼性を有するRBUDPを実装し、TCPとの比較を行う。RBUDPはUDPを用いてデータ転送を行い、送信側で全てのデータを送信した後、終了のシグナルを受信側へと送信する。受信側はそのシグナルを受け取ると、損失したパケットの情報を伴ったACKを返し、それを受け取った送信側は再送処理を行い、全てのパケットが受信側へと到達するまで処理が繰り返される(図1)。このように、RBUDPは高速性と信頼性を確保している。

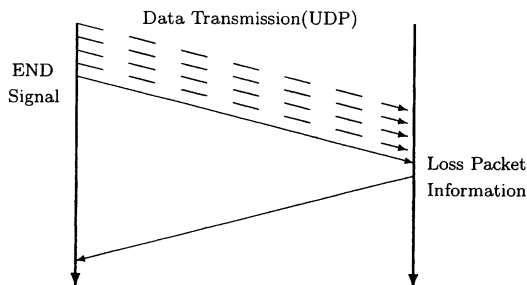


図 1: RBUDP

### 3 遠隔可視化処理におけるデータ転送実験

#### 3.1 実験環境

本実験では、東北大学と京都大学間をネットワークで介した遠隔可視化環境を構築した。図 2 に示すように、東北大学にサーバとして、スカラ型並列コンピュータ TX-7/AzusA を、京都大学に可視化表示用クラスター (以下 VG Cluster) をクライアントとして配置している。また長距離ネットワーク網として、SuperSINET(実験用専用回線) を介して実験を行う。今回実験を行う環境下でのサーバ/クライアント間で計測された RTT(Round-Trip Time) は平均 28[msec] である。

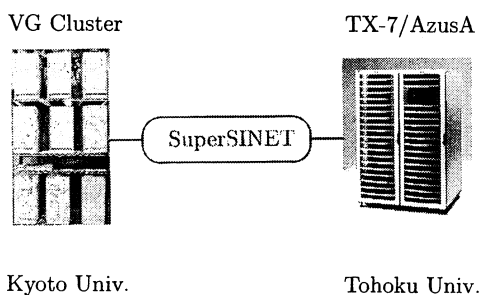


図 2: 実験システム構成

実験で用いるサーバ、クライアントの概要を、それぞれ表 1, 2 に示す。なお、クライアントの VG Cluster は、1 台をマスター (表示用)、8 台を可視化処理用としたクラスター PC を形成し、分割処理を行う。

また、VG Cluster でレンダリングされた画像の重

表 1: 並列コンピュータ (サーバ) 仕様

TX-7 / AsusA	
CPU	Intel Itanium 800 MHz 4 MB L3 cache
PE	16
Memory	32 GB / 16 GB
OS	RedHat Linux

表 2: VG Cluster (クライアント) 仕様

VG Cluster (表示用 1 台, 可視化処理用 8 台)	
CPU	Inter Pentium4 2.4 GHz
Memory	1 GB DDR SDRAM PC2100
GPU	nVIDIA GeForce FX5950 Ultra (256MB)
OS	RedHat Linux 7.3 (2.4.18-3SCORE)

畳を行う 2 次元画像重畳装置には、MPC(Mitsubishi Precision Co.,Ltd) 製の Image Compositor を使用する (図 3)[4]。本装置は、BSP(Binary Space Partitioning) 手法を用いたアルファブレンディング専用ハードウェアともいえる。更に、2 出力 (Dual 4:1) 又は 4 出力 (Quad 2:1) 装置として利用可能であるため、大型の多面体スクリーンシステムにも容易に対応可能とされる。



図 3: Image Compositor

#### 3.2 実験内容

サーバ/クライアント環境での遠隔可視化処理は、図 4 に示すように分類され、本研究で我々が構築した遠隔可視化システムは、図 4 の C のモデルに相当する。サーバでは可視化対象である三次元ボリュームデータの読み込みや、クライアントとのシステム情報、データ情報の交換が行われる。その後クライアントの台数に応じてデータを分割し、クライアント側の各レンダリングノード (vg01~vg08) へ送信する。本実験においては、可視化データは図 5 のように x, y, z 方向にそれぞれ等分し、さいの目のように計

Data Storage	Data Calculation	Geometric Calculation	Rendering	Display
--------------	------------------	-----------------------	-----------	---------

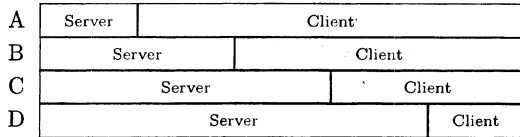


図 4: 遠隔可視化処理モデル

8個に分割されている。分割されたデータは各可視化ノード (vg01~vg08) にて受信され、それぞれレンダリング処理が施される。その後、Image Compositorにて重畳が行われ、表示用 PC で最終画像を得ることになる。その処理の様子を、図 6 に示す。

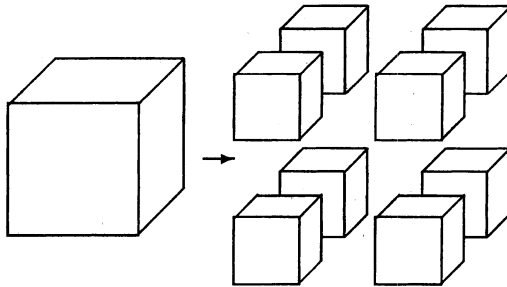


図 5: データ分割方法

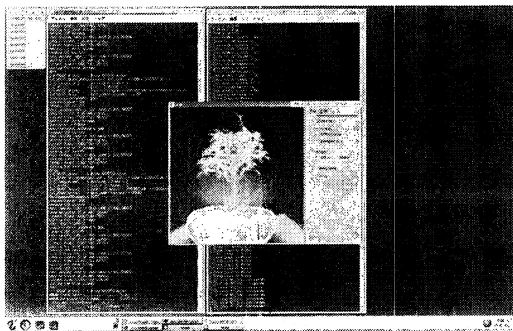


図 6: 最終画像が表示された表示用 PC 画面例

今回用いる可視化対象ボリュームデータの格子数、データサイズを表 3 に、実際に得られた重畳画像を図 7-9 に示す。また本実験では、トランスポートプロトコルに TCP と RBUDP を用いた場合の遠隔可

視化処理を行い、それぞれの場合のネットワーク転送時間を測定した。また、RBUDP では送信レートを変化させて、それぞれ転送時間と共にパケットロス率を測定し、転送時間への影響を調査した。

表 3: 可視化対象データ

データ名	格子数	データサイズ [byte]
Hydrogen	64 × 64 × 64	262144
Aorta	128 × 128 × 128	2097152
Bonsai	256 × 256 × 256	16777216

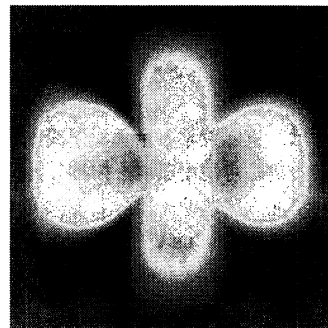


図 7: Hydrogen (64 × 64 × 64)

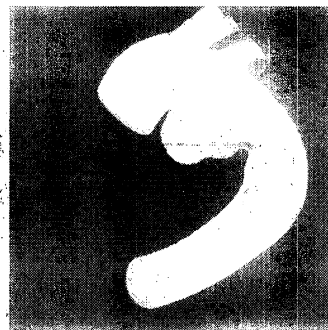


図 8: Aorta (128 × 128 × 128)

### 3.3 実験結果

実験によって得られた可視化処理におけるデータ転送時間を、図 10 に示す。これより、Aorta や Bonsai において、送信レートを 100 [Mbps]、200 [Mbps] に設

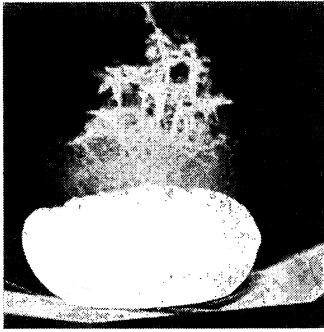


図 9: Bonsai (256 × 256 × 256)

定した RBUDP を用いた場合、データ転送時間の改善がみられた。しかし、送信レートを 300 [Mbps] とした RBUDP を使用した場合、TCP を用いた時よりもデータ転送にかかる時間が大きくなり、200 [Mbps] の時は TCP よりは良い結果を出したものの、100 [Mbps] の時よりも時間がかかっていることが分かる。

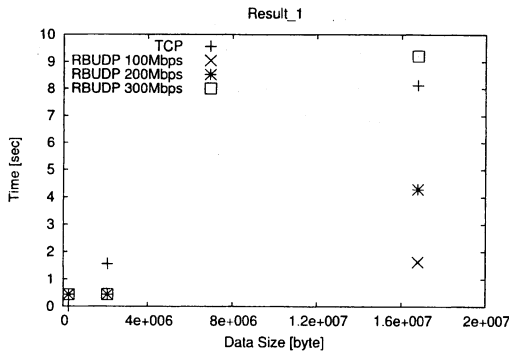


図 10: データサイズ毎のネットワーク転送時間

次に、RBUDP の送信レートを変化させた際のパケットロス率を測定した結果を、図 11 に示す。

図 11 より、100 [Mbps] ではパケットロスは発生していないが、200 [Mbps], 300 [Mbps] では、Bonsai を可視化する際に大量のパケットロスが発生しているのが確認できる。

## 4 考察

図 10 より、TCP と送信レート 100 [Mbps] に設定した RBUDP を比較すると、データサイズが増加するにつれて RBUDP によるデータ転送の有効性が高

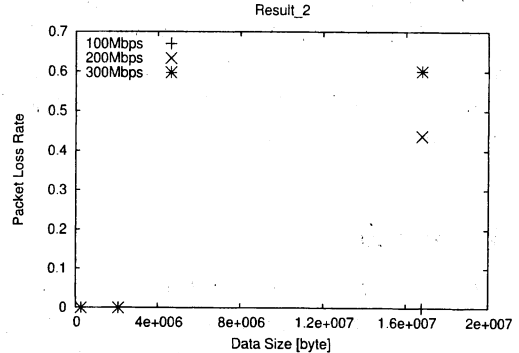


図 11: データサイズ毎のパケットロス率

まると言える。Bonsai を可視化した際、データ転送時間は TCP が 8.136 [sec]、送信レート 100 [Mbps] の RBUDP が 1.6355 [sec] と、RBUDP を用いることで大幅に改善されたことが確認できる。データサイズとデータ転送時間より、Bonsai を可視化したときのそれぞれのスループットを求めると、表 4 となる。

表 4: スループット (Bonsai)

使用プロトコル	スループット [Mbps]
TCP	16.50
RBUDP 100 Mbps	82.07
RBUDP 200 Mbps	31.26
RBUDP 300 Mbps	14.55

TCP では、ウィンドウ制御によりウィンドウサイズを変化させ、ネットワークの効率的な利用が図られている。そこで、表 4 と RTT = 28 [msec] より TCP のウィンドウサイズを求めると、 $5.77 \times 10^4$  [octet] であり、仮に送信レート 100 [Mbps] の RBUDP のスループットを TCP で実現しようとする、 $2.87 \times 10^5$  [octet] のウィンドウサイズが求められることになる。ウィンドウサイズは通常 65535 [octet] が上限であるため、これを達成するにはウィンドウスケリングの操作が必要となる。

また、RBUDP において送信レートを上げてパケットロスを発生させると、TCP 以上にデータ転送に時間を要する結果になった。これは、RBUDP は輻輳制御が行われないため、輻輳が発生するとネットワークの利用効率が極端に低下するためと思われる。よって、RBUDP には帯域を考慮した、適切な送信レートの決定が肝要である。また、今回は専用線を用い

た実験であるため、他のトラフィックを考慮する必要がないが、RBUDPは送信レートを動的に変化させることができない。そのため、他の通信による影響を大きく受ける環境下における対応方法も今後検討する必要があると考える。

## 5 まとめ

本稿では、遠隔可視化処理の高速化を目的として、ネットワークにおける伝送遅延の削減のため、RBUDPを遠隔可視化システムに実装し、性能評価実験を行った。その結果、帯域を考慮した送信レートでのRBUDPの使用により伝送遅延の削減を確認した一方、輻輳発生時におけるRBUDPの問題点も明らかになった。

データ転送における高速性と信頼性を追及した研究は、RBUDPに限らず様々な事例が存在する[5]。今回用いたRBUDPのようにUDPベースでデータ転送を行い、送信レートを制御するSABUL(Simple Available Bandwidth Utilization Library)[6]や、TCPを用いて、ウィンドウサイズの動的な決定や、接続の並列化により高速化を図るGridFTP[7]などの取り組みもある。そのような試みや遠隔可視化の特性をより深く調査し、遠隔可視化の更なる高速化を目指した検討を進める予定である。また、今回計算で算出したウィンドウサイズにおいて、実際にTCPを動作させて遠隔可視化を行い、その振る舞いを調査することも今後の課題と考える。

## 参考文献

[1] Y.Ebara et al. "Remote Visualization using Resource Monitoring Technique for Volume Rendering of Large Datasets", the 2004 International Symposium on Applications and the Internet (SAINT 2004), pp.309-312, 2004.

[2] Jorji Nonaka et al. "Remote Visualization Framework for Promoting Cooperative Visualization" The Fourth International Symposium on Advanced Fluid Information and The First International Symposium on Transdisciplinary Fluid Integration 2004 (AFI/TFI2004), pp.165-170, 2004.

[3] Eric He et al. "QUANTA: A Toolkit for High Performance Data Delivery over Photonic Networks" Journal of Future Generation Computer Systems (FGCS), Elsevier Science Press, Volume 19, Issue 6, pp.919-934, August 2003.

[4] MITSUBISHI PRECISION CO.,LTD.  
ボリュームグラフィッククラスタシステム  
[http://www.mpcnet.co.jp/product/vg/index\\_vg.html](http://www.mpcnet.co.jp/product/vg/index_vg.html)

[5] L. Renambot et al. "Griz: Experience with Remote Visualization over an Optical Grid" Journal of Future Generation Computer Systems (FGCS), Elsevier Science Press, Volume 19, Issue 6, pp.871-882, August 2003.

[6] Yunhong Gu, Robert Grossman "SABUL: A Transport Protocol for Grid Computing" Journal of Grid Computing, Volume 1, pp.377-386, 2003.

[7] Bill Allcock et al. Sam Meder, Veronika Nefedova, Darcy Quesnel, Steven Tuecke "Data Management and Transfer in High-Performance Computational Grid Environments" Parallel Computing 28, pp.749-771, May 2002.