

# 仮想環境を用いた 8K 映像ストリームサーバの実現

瀬林克啓<sup>1</sup> 岩崎昂大<sup>1</sup> 池田哲也<sup>1</sup> 丸山充<sup>1</sup>  
君山博之<sup>2</sup> 青木弘太<sup>3</sup> 栗本崇<sup>4</sup> 漆谷重雄<sup>4</sup>

**概要:** ネットワークの広帯域化に伴い、8K (7680×4320 画素)映像は、細部まで精密に映像化できることから、医療・教育・放送分野での期待は高い。そこで我々は、クラウド上の空きリソースで安価に構成した 8K 映像サーバ仮想マシン(VM)を複数連携動作させ、トータルでビデオ伝送レート 24Gbps の 8K 非圧縮映像の伝送・蓄積を可能とする 8K 映像ストリームサーバの実現を目指している。

我々は、SINET5 の NFV 環境において、物理サーバ 1 台あたり 2 台動作させた 8K 映像サーバ VM を 8 台連携させた 8K 映像ストリームサーバで、トータルでビデオ伝送レート 24Gbps の 8K 非圧縮映像伝送の安定動作を実現した。また、SINET5 の 3 拠点の NFV に計 10 台の 8K 映像サーバ VM と仮想 IP マルチキャストルータを配置し、帯域浪費の抑制するとともに、10 台の 8K 映像サーバ VM の中から任意の 8 台を選択して構成した 8K 映像ストリームサーバで 8K 映像を安定送信するとともに、その 8 台を切り替える 8K 映像ストリームサーバの高度化実験にも成功した。

本稿では、SINET5 の NFV 環境において安定動作する 8K 映像サーバ VM の構成 (CPU 数及びメモリ量) とカーネルパラメータの導出方法と 8K 映像ストリームサーバの高度化について説明する。

## Implementation method for 8K video stream server using virtual environment

KATSUHIRO SEBAYASHI<sup>1</sup> KOUTA IWAZAKI<sup>1</sup> TETSUYA IKEDA<sup>1</sup>  
MITSURU MARUYAMA<sup>1</sup> HIROYUKI KIMIYAMA<sup>2</sup> AOKI KOUTA<sup>3</sup>  
TAKASHI KURIMOTO<sup>4</sup> SHIGEO URUSHIDANI<sup>4</sup>

### 1. はじめに

ネットワークの広帯域化に伴い、超高精細の映像ストリームを扱うアプリケーションの普及が見込まれている。特に 8K (7680×4320 画素)映像は、細部まで精密に映像化できることから、医療・教育・放送分野での期待は高い。このような 8K 映像にクラウドやエッジコンピューティングを用いて可視化した映像等を合成することで、映像ストリームアプリケーションに新たな展開が期待できる。

しかしながら、8K 映像のデータ量は 24Gbps から 144Gbps ときわめて大きく、伝送・蓄積・処理には、ネットワーク帯域の確保に加え、単体ノードとして高いレベルでのパケット処理性能、並列演算性能、蓄積配信性能が求められる。そこで我々は、8K 超高精細映像の伝送・蓄積・処理機能を NFV(Network Function Virtualization)基盤を持つクラウド内のデータセンタ(DC)やクラウド・エッジに分散配置し、それらを自在に連携させ必要に応じた映像ワークフローを自動構成する映像処理プラットフォームの確立を目指している[1][2]。この映像処理プラットフォームの伝送・蓄積機能として、我々は、クラウド上の空きリソースで構成した仮想マシン(VM)を用いて安価に構成可能な 8K 映像ストリームサーバの実現を目指している。この 8K 映像ストリーム

サーバは、8K 映像サーバ VM を複数連携動作させ、トータルでビデオ伝送レート 24Gbps の 8K 非圧縮映像の伝送・蓄積を可能とする。この 8K 映像ストリームサーバは、これまでの Interop Tokyo 2018 での実証実験[3]において 8K 映像サーバ VM の安定性と DC 内のトラヒックの最適化の課題、複数の DC に跨る 8K 映像サーバ VM の配置のシステム化の必要性が明らかになっていたが、これらの課題を解決し、2022 年 2 月に NICT 超高精細映像を用いた広域映像配信実証実験において実現性を確認した。

本稿では、これらの課題の解決方法を提案し、その実現性を検証する。まず、8K 映像ストリームサーバ上記の課題について説明する。次に、各課題の解決方法について説明し、更に、第 4 章では 2022 年 2 月に NICT が主催した超高精細映像を用いた広域映像配信実証実験で実施した実証実験における検証結果について述べる。最後にまとめと今後の課題を述べる。

### 2. 8K 映像ストリームサーバの概要と課題

本章では、8K 映像ストリームサーバの概要と課題について説明する。

#### 2.1 8K 映像ストリームサーバの概要

8K 映像ストリームサーバは、8 台の 8K 映像サーバ VM

1 神奈川工科大学  
Kanagawa Institute of Technology

2 大同大学  
Daido University

3 ミハル通信株式会社

Miharu Communications Inc.

4 国立情報学研究所  
National Institute of Informatics

で映像ストレージで構成される（図 1）。

各 8K 映像サーバ VM（図 2）は、8K-DG 方式で 8 分割された 8K 映像データの 1 つをストレージから読み出し、3Gbps の映像ストリームとしてユニキャスト又はマルチキャストで IP 網に送信する。受信拠点では 4 台の IP-SDI 変換装置を RS 信号と SDI Genlock 信号で連携させており、各 IP-SDI 変換装置は 2 台の 8K 映像サーバ VM から映像ストリームを受信し、映像データを 3G-SDI 信号に変換する。この 3G-SDI 信号は HDMI 信号に変換され、8K ディスプレイに送出される。

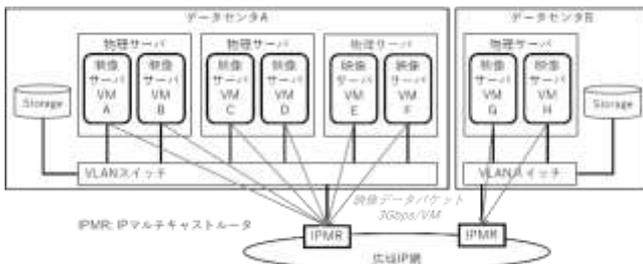


図 1 8K 映像ストリームサーバの構成

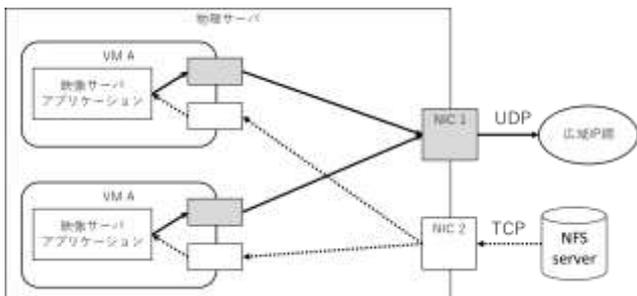


図 2 8K 映像サーバ VM の構成

## 2.2 8K 映像ストリームサーバの課題

現在の 8K 映像ストリームサーバと 8K 映像サーバ VM には下記の課題がある。

**課題 1:** 8K 映像サーバ VM の安定動作のための適切な VM リソース配分及びカーネルパラメータのチューニング。

Interop Tokyo 2018 での実証実験[3]において、8K 映像サーバ VM 単体の性能が安定せず、VM のカーネルパラメータの `net.core.rmem_max` と `net.core.wmem_max` を増加させたところ状況が改善したが、継続したノイズのない正常な映像伝送ができたのは数分間のみであった。

**課題 2:** IP マルチキャストパケットによる DC 内の帯域の浪費の抑制。

現在の 8K 映像ストリームサーバの構成は、DC 内の 8K 映像サーバ VM を、VLAN で直接 DC 外の広域 IP マルチキャスト網の IP マルチキャストルータ(IPMR)に接続している。通常、送信ホストと隣接マルチキャストルータ間では枝刈りが行われない。そのため、映像サーバ VM がマルチキャストグループ宛に送信した 3Gbps 映像データ (IP パケットで 3.2Gbps) のマルチキャストパケットは、たとえ受信者が存在しない場合でも広域 IP マルチキャスト網の IPMR に送信され、DC 内の同一リンク上に VLAN 多重された他

ユーザの使用帯域を圧迫する。

**課題 3:** 各拠点のリソース使用状況に応じて 8K 映像サーバ VM を配置するシステム化。

DC 内のリソースは他のユーザとの共有となるため、他ユーザの利用状況にとっては、映像サーバ構成に必要なサーバリソースとネットワークリソースの確保が困難となる。そのため、各 DC 内のリソース使用状況に応じて代替の 8K 映像サーバ VM を配置し、リソース不足になった 8K 映像サーバ VM のストリーム送信を引き継ぐシステム化が必要になる。

課題 1 については、3.VM 性能の基本評価で安定動作の原因を特定し、4. VM 入出力性能のチューニング手法の検討でピーク性能向上に寄与するチューニングとシステム全体の性能安定化に寄与するチューニングと選別したチューニングについて検討し、最終的なシステム構成とチューニングを明らかにする。

課題 2 と課題 3 については、5. 8K 映像ストリームサーバ高度化の提案でソフトウェア IPMR の導入によるフラッディングトラフィックの局所化と DC リソースの状況に応じて 8K 映像サーバ VM を再配置するシステム化を提案し、6. 実現性の検証では、今回明らかにしたパラメータで 8K 映像サーバ VM を構成し、5. で提案する構成で実施した評価実験について説明する。

## 3. 8K 映像サーバ VM の基本性能評価

前項の課題 1 の解決のため、8K 映像サーバ VM の入出力性能の不足の原因について、ベンチマークアプリケーションと映像再生アプリケーションを併用して、1) 蓄積読出し処理、2) 回線配信処理、3) 複数 VM の同期処理の観点で VM 単体の基本性能試験を行った。このうち 1) 蓄積読出し処理の試験結果を考察する。

### 3.1 実験環境

実験環境全体図を図 3 に示す。SINET の北海道 NFV の物理サーバ 3 台と東京 NFV の物理サーバ 1 台を使用し、8K 非圧縮映像伝送に必要な 8VM を動作させる。

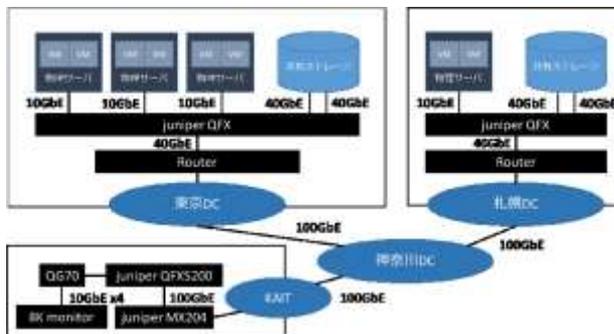


図 3 実験環境の構成

物理サーバのスペックは、CPU が Intel(R) Xeon(R) CPU E5-2620 v4 @2.10GHz、NIC は Intel 82599ES 10-Gigabit SFI/SFP+, OS は Ubuntu 16.04 であり、VM のスペック

は、vCPU core 数を 10 (完全仮想化)、メモリ 50GB、OS には Ubuntu16.04 を用いた。

### 3.2 システム基本性能試験

#### 3.2.1 蓄積読出し処理性能試験

本性能評価実験では、指定したファイルをマシンの限界性能で読み込み、受信側インタフェースでの受信フレーム数や指定フレームの受信に要した平均時間、合計時間等を計測し、ログ出力するファイル読出しベンチマークプログラムを用いて VM、物理サーバの蓄積読出し性能を測定した。

実験では北海道 NFV 及び東京 NFV の共有ストレージから、NFSv4 を用いて実際のアプリケーションで用いる 5 分程度の 8K 非圧縮映像のストリームを読出した。

北海道 NFV 内のある 1 台目の物理サーバ (北海道物理サーバ 1) の 2VM (VM1 と VM2) から、共有ストレージ内の映像を同時に読出した際には、多くの場合、規定の 60 フレームを 1 秒以内に読出しているが、VM1 では最長で 1.10 秒、VM2 では最長で 1.57 秒を要することがあった。ここから、要求性能である 1VM あたり 3Gbps を多くの場合満たしているものの、時折性能が低下していることが確認できた。北海道 NFV の 2 台の VM からの読出し性能を図 4 に、東京 NFV の 2 台の VM からの読出し性能を図 5 に示す。縦軸が伝送速度 (Gbps)、横軸が秒間を表している。また VM の vCPU core 数を 10 core とし、CPU-pinning、vlan offload、NIC のリングバッファの拡張の有無の組合せを変え、計 8 通りの条件について測定している。



図 4 北海道物理サーバ 3-VM1 の映像読出し性能測定結果



図 5 北海道物理サーバ 3-VM2 の映像読出し性能測定結果  
この結果、全ての VM において読出し能力は 3Gbps から 7Gbps 付近を推移しており、どの設定の組み合わせでも要

求性能である平均 3.2Gbps 以上のスループットを記録した (0.2Gbps は映像伝送時のオーバーヘッドである)。しかし伝送性能の平均値が最大となる設定については各 VM で一致しなかったため、分散が少なく安定した設定の組み合わせを VM 毎に適用する手法が最も有効であると考察した。またこの実験によって、今回の VM スペックで、読出し処理のみであれば同一物理ホスト上の 2VM の並列稼働が性能として問題ないことが明らかになった。しかし、2.1 の評価では、アプリケーションを複数 VM で動作させると性能が不安定になったため、この内容と矛盾する結果となった。そこで読出し処理と配信処理を並列化させた際に VM の処理能力が不足するのではないかと推測した。

#### 3.2.2 回線配信処理性能試験

映像配信を模擬したダミーデータを一定レートで送信し続けるツール (test\_sender) を用いて、VM の配信性能を確認した。まず VM で本ツールのみを実行させ、送出性能に問題がないか確認した。次に前項で推測した、読出し処理と配信処理の並列化による VM の性能不足が起きるかを確認した。

test\_sender で各物理サーバのうち 1VM で 6.4Gbps (2 ストリーム分) のダミーデータを送信し、linux の dstat コマンドを用いて実際の送信レートを観測結果、配信処理のみであれば 1VM で要求性能の倍である 6.4Gbps であっても問題なく安定して出力できることが判明した。

また VM で読出し処理と配信処理を並列化させた際の dstat コマンドの出力結果では読出し性能が 1~3Gbps の間を強い揺らぎで推移しており、配信性能においても複数の VM では所々強い揺らぎが発生した。従って、現構成の VM では読出しと配信を並列化させると性能が不安定になることが判明した。

#### 3.2.3 複数 VM の同期処理

サーバアプリケーションを映像伝送時と同様に複数 VM で動作させ、VM 間の同期ずれや特定の VM の伝送性能の低下、送受信性能の低下の程度を確認した。図 6、図 7 は、4VM、8VM で 8 ストリームを送信時、アプリケーションが保持するフレームのバッファ量の推移と、送受信レートをまとめたグラフである。



図 6 4VM で 8 ストリーム送信の伝送性能



図 7 8VM で 8 ストリームを送信時の伝送性能

ここではアプリケーション側で最大 100 フレームまでバッファリング可能と設定しており、読出し性能が充分ある正常時の場合、その最大数まで映像フレーム(青線の write wait 値)がメモリに格納される。

しかしここでは 4VM の場合も 8VM の場合も、共有ストレージからの読出し性能(青線)に揺らぎがあり安定しておらず、加えて図 6 右下の北海道 VM2 においては、2 ストリーム中 1 ストリームが欠落し、伝送レートは 3.2Gbps であった。送信レートは一部を除き安定していたものの、これはアプリケーションが最後に受信したフレーム番号の映像を繰り返し送出する仕様である為であり、実際の映像は同期が外れ色情報の欠落が起きていた。

また、本項の実験では読出し性能不足から、多くの場合アプリケーションが用意したバッファに映像フレームが蓄え切れておらず、empty の値が上昇する結果となった。

従って、本構成の VM を使用してアプリケーションを動作させた場合、共有ストレージからの映像読出し性能が不足し、慢性的にアプリケーションが用意したフレームのバッファが枯渇する現象があること、加えて、これにより複数の VM の映像同期が乱れ、受信拠点側で色情報の欠落した映像が出力されることが判明した。

### 3.2.4 システム基本性能試験の結果

ベンチマークアプリケーションとダミーデータ送信ツールを使用して VM の基本性能を測定し、不足性能を明らかにした。また、これにより正常な映像伝送ができない理由の手掛かりが明らかとなった。具体的には、

1. VM の映像読出しと配信は単体動作であれば性能に問題は無いが、併用時に性能が不足する。
2. 1 によってアプリケーションが用意したフレームバッファが慢性的に枯渇する。
3. 複数 VM が同時に動作する場合、ストリームの欠落や読出し性能の揺らぎが発生する
4. 2 により複数 VM 間の同期が乱れ、受信拠点側で色情報の欠落等が表出する

これらを踏まえ、VM の単体構成やシステム全体に用いる技術要素を再度検討し、要求性能を適切に引き出すチューニングを様々な観点から推測して実施した。

## 4. VM 入出力性能のチューニング手法の検討

基本評価で明らかにした VM の入出力性能の不足に対し、様々な観点から有効と思われるシステムチューニングを検討する。ここで検討したものを表 1 に示す。

表 1 検討したチューニング方法

最新のNICを活用	SR-IOV	NIC offload
vCPU割当制御	CPU-pinning	RPS
メモリ割当制御	txqueuelen	Kernel parameter
	APL buffer pages	Flow control
	Hugepage	
基本構成要素の選択	vCPU core数	VM memory
	VM OS	Virtio
NWオーバーヘッド削減	Video packet size	Storage controller
	Ramdisk	
ファイルシステム	NFS	GPFS

### 4.1 ピーク性能向上に寄与するチューニング

チューニング項目の中で、SR-IOV の有効化、NIC offload (CentOS)の有効化、CPU pinning の実施、カーネルパラメータの変更、Video Packet Size の増加、Txqueuelen の増加、Kernel Param Mem の増加、映像サーバの buffer pages の増加は、VM のピーク性能向上に寄与した。以下、特に効果が高かった項目について説明する。

#### 4.1.1 SR-IOV

SR-IOV(Single Root I/O Virtualization)は NIC に搭載されている機能で、PCI デバイスを多重化して仮想インタフェースにバイパスする技術である。これを用いて NIC 仮想化におけるオーバーヘッドを減少させ、共有ストレージからの読出し性能に変化があるか検証した。

ストレージ側ネットワークに接続される物理 NIC と配信側ネットワークに接続される物理 NIC のそれぞれを SR-IOV 機能を用いて仮想的に分割し、各物理 NIC あたり VF(virtual function)を 2 つ作成する。この VF を 1 つずつ VM の仮想インタフェースに対応させることにより、物理 NIC 内では 2VM のトラフィックが競合しない状態が実現できる。

そこで、北海道、東京 NFV 上の 8VM 全てにこの SR-IOV を適用した結果、全ての送信側インタフェースの伝送レートが 382MB/s(3Gbps)で安定することが確認できた。

SR-IOV の適用により、送信側インタフェースの伝送レートは SR-IOV 適用前に比べて格段に安定したが、受信側インタフェースの伝送レートについては安定化が見られず、この時点で共有ストレージの読出し性能不足には別の原因が存在することが明らかとなった。

#### 4.1.2 CPU-pinning

CPU-pinning は VM、あるいはプロセス単位で特定の CPU コアのみを使用する技術である。複数の VM が同一の物理サーバ上で稼働し、かつ大容量ストリーミングのように長時間 CPU に負荷が生じるプロセスを動作させる際には、使用する CPU コアが処理中に無秩序に移動すると CPU リソースの競合が発生する可能性があり、オーバーヘッドになり得る。そのため、本技術を用いて VM 毎に使用する vCPU

を固定する必要があるのではないかと推測した。

Pinningにあたっては、同一 NUMA ノード上の CPU コアに 1 台の VM の vCPU コアを全て揃えることにより、リモートメモリにアクセスする可能性を減らすことができる。これによって VM 間の vCPU リソースの競合防止に加え、CPU のメモリアクセスにかかるオーバーヘッドを減少させることが可能になる。本検証では vCPU 4core を VM 毎に番号順に割当て、それぞれのコアが同一 NUMA ノード上に配置されるような設定を実施した。これにより、共有ストレージからの読出し性能の安定化、特定の CPU コア使用率の上昇防止等の効果が得られた。

## 4.2 システム全体の性能安定化に寄与するチューニング

チューニング項目のなかで、VM の OS 変更、vCPU core 数の削減、VM のメモリ量の削減、リモートファイルシステムの選択、RPS(receive packet steering)の適用がシステム全体の性能安定化に寄与した。以下、特に効果のあった項目について説明する。

### 4.2.1 VM の OS の検討

基本評価実験では、物理サーバ、VM 共に OS は Ubuntu16.04 を使用し、vCPU core 数を各 VM に 8 つ以上割当てた基本構成を採用していた。

#### 4.2.1.1. Ubuntu の再評価

これまで共有ストレージからの映像読出し性能が不足しており、システムで 8VM を動作させた際には、性能が不足したいくつかの VM においてバッファリングフレームの枯渇が発生していた。これが複数 VM 間の同期ずれを引き起こし、受信拠点で出力される映像ではノイズが散見されていた。

上記の現象が VM の OS 変更によってどの程度解消されるかを検証すべく、まず Ubuntu を用いた VM の基本性能と現段階で判明している現象を整理した。

1. 読出し処理、配信処理を単体で動作させた場合には要求性能を満たしている。
2. 2 つの処理が重なると、送信性能は 1VM あたり 3Gbps を保てるが、読出し性能は安定しない。
3. NIC に SR-IOV を適用しても単体の読出し性能自体が不足している。
4. 1 台の物理サーバに 2VM を稼働させても 1VM までしか安定稼働しない。

次に複数のリソースモニタリングコマンドを用いて、映像伝送中の Ubuntu VM の挙動の特徴を再観測した。

その結果、時折ランダムに 1~2 台程度、VM の映像読出しに用いる vCPU core の使用率が 100% 付近まで上昇する現象を発見した。加えてこの時 netstat -s コマンドの出力を確認すると、VM と共有ストレージの間で再送要求(segment retransmitted)が増加していた。これにより、VM の読出し性能不足の原因は共有ストレージへのパケット再送の頻発であることが明らかになった。本事象の発生時にはストレージ

からのフレーム読出しが遅れ、それに伴いアプリケーションバッファが枯渇し、複数 VM 間での映像同期が外れ出力にノイズや色情報の欠落が発生する。

また、Ubuntu には I/O に負荷がかかると特定の CPU リソースが過負荷となり、そのコアを使用できなくなる不具合があることが同時期に判明した。本事象は Linux Kernel 2.6 から存在しており、カーネルスレッドの一つである kworker の誤作動に起因するものである。ベンチマークプログラムの動作には問題はないが、サーバアプリケーションを使用すると当該事象が引き起こされる可能性があることが判明した。

#### 4.2.1.2. CentOS の評価

次に、Ubuntu の再評価で得た情報を基に、StarBED 上の物理サーバ群を用いた過去の研究において、安定的にアプリケーションが動作した実績がある CentOS 6.9 の VM を構築して同様にアプリケーションを動作させ、OS 間で挙動を比較検証した。

アプリケーション動作による伝送性能を dstat コマンドで確認すると、読出し性能は 2.5~5Gbps と安定していないが、同時刻に読出した量の映像フレームレートをほぼ全て送出しており、配信性能や CPU リソースの圧迫可能性の低さにおいて Ubuntu を上回った。ここから読出し性能の安定化に向けて他項で説明するパラメータチューニングを施すことにより、この VM の性能は実用可能な性能に改善された。

なお、本研究で用いた物理サーバでは Ubuntu 16.04 を使用しているが、ホスト OS 側においては CPU リソースの圧迫や読出し性能の不足は見受けられなかった。このことから、VM 上で Ubuntu を使用した場合、プロセススケジューラの挙動に何らかの問題が発生し得るのではないかと考えられる。

### 4.2.2 vCPU core 数の検討

プロセススケジューラの挙動に影響を与えるパラメータの一つに VM の vCPU core 数がある。OS に Ubuntu を使用して kworker の誤作動を確認した際、VM の vCPU core 数は 8core であった。そのため、ここでは過度なリソース割当てが性能低下を引き起こす可能性を考え、前項の実験によって本システムの全ての VM を CentOS 6.9 に変更した後、vCPU core 数の減少について検討した。

vCPU core 数を 8core とした場合と、4core にした場合で、共有ストレージから 8VM の同時読出し性能を計測した結果、VM の vCPU core 数は 8core の場合に比べて 4core のほうが読出し性能が飛躍的に安定する事が明らかとなった(図 9)。このことから、VM を用いた大容量ストリーミングにおいては使用する vCPU core を最小限まで減らし、高速伝送時に性能劣化が起きないよう仮想化のオーバーヘッドを減らすことが有効であると考察した。

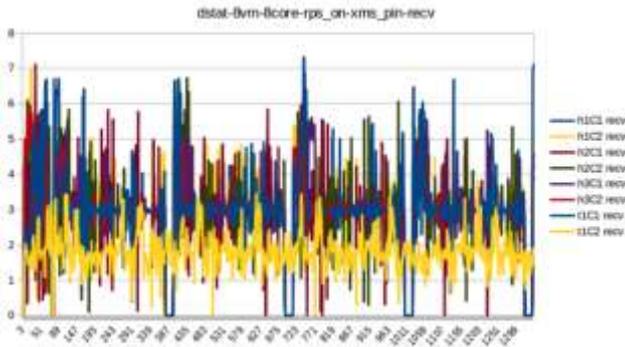


図 8 vCPU 8core で動作時の VM 毎の伝送レート

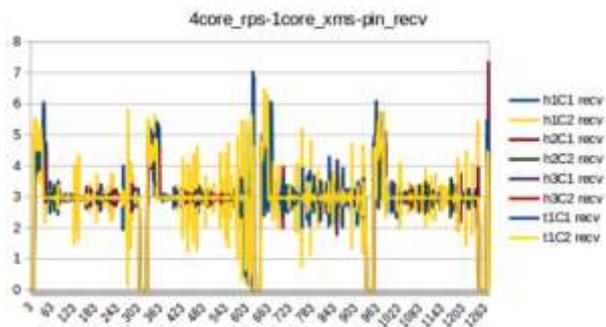


図 9 vCPU 4core で動作時の VM 毎の伝送レート

#### 4.2.3 VM メモリ保有量の軽量化

前項にて vCPU core 数の削減が読出し性能の安定化に有効であったことから、同様にメモリ保有量の軽量化についても検討した。

物理サーバに近い構成として VM でもメモリを多く確保していたが、アプリケーションの動作に必要なメモリ量を調査すると、最低 8GB を確保していれば動作可能であることが判明した。この為、1VM が保有するメモリ量を 50GB から 32GB、16GB に削減して読出し性能の比較を行った。

比較方法は、アプリケーションが用意したフレームバッファの中に現在入っているフレームの番号が記録されているファイルを、watch コマンドを用いて毎秒観測し、VM で読出し性能の不足によるバッファリングフレームの枯渇が起こらないか確認した。その結果、32GB の場合も 16GB の場合も枯渇は起こらず、全ての VM において正常にバッファリング機構が作動した。

なお、1 台の VM にメモリを 50GB 確保していた際には読出し性能が追い付かず、バッファリングフレームの枯渇が時折見受けられていた。これらのことから、多量のメモリを VM に割当てた場合、リモートメモリを確保する確率が高まり大容量伝送を行う際にリモートメモリへのアクセスによるオーバーヘッドが増加し、結果として性能低下が引き起こされているのではないかと推測される。

#### 4.2.4 評価結果

以上、8K 非圧縮映像配信サーバの実装における VM の入出力性能の不足に対し、対策となり得る具体的なチューニング手法を検討してきた。これらを考察すると、本シ

テムに有効なチューニング手法は、VM のピーク性能向上に寄与するものと安定度の向上に寄与するものに分類できることが明らかとなった。加えて、本システムにおけるファイル読み込みの特性や仮想化のオーバーヘッドの影響等により、今回検討するも効果を得られなかったチューニング手法も数点見つかった。実験結果を、表 2 チューニング項目の効果の評価結果に示す。

この表中の効果の欄に◎を記した項目は、特に有効であったチューニングであり、ピーク性能の向上に寄与した SR-IOV と CPU-pinning の有効化、安定度の向上に寄与した VM の OS 変更、VM 保有メモリ量の軽量化、vCPU core 数の削減の 5 つである。

RPS の効果は送信インタフェースに限定され、ストレージコントローラの割り振りは、ストレージの使用状況に依存するため△とした。

表 2 チューニング項目の効果の評価結果

チューニング項目		効果	
サーバ	ピーク性能の向上に寄与	SR-IOVを有効化	◎
		NIC offload (CentOS)	○
		CPU pinning	◎
		Video Packet Size増加 ※1	○
		TxqueueLen 増加	○
		Kernel Param Mem 増加	○
	安定度の向上に寄与	XMS buffer pages	○
		VM CPU core 削減	◎
		VM Mem 削減	◎
		VM OS 変更(CentOS)	◎
その他	Receive Packet Steering ※2	△	
	ストレージコントローラの割り振り	△	
共有ストレージ	SW flow control	○	
	NFS	○	
	GPFS	×	
※1 受信拠点側も要調整			
※2 softirqには効果有			

VM を用いた 8K 非圧縮サーバの実現

評価実験として、これまでに有効性が認められたチューニング手法を 8K 非圧縮映像配信システムに適用し、改めて神奈川工科大学へ向けて非圧縮 8K 映像配信実験を行った。最終的なシステム構成とチューニングについて概説し、次にこれを用いた映像配信実験について述べる。

#### 4.3 最終的なシステム構成とチューニング

これまで行ってきた様々なチューニングにより、最終的なシステム全体構成は表 3 のようになった。

表 3 最終構成のチューニング一覧

	検討事項	検討前	検討後
サーバ	SR-IOV	off	on
	VM OS	Ubuntu18.04	CentOS8.9
	vCPU core	8以上	4
	VM memory	50GB	18GB
その他	Kernel parameter	default	表5に記載
	CPU pinning	off	on
	Video packet size	8000 byte	8800 byte
	TxqueueLen	2000 byte	3000 byte
	XMS buffer pages	150 frame	600 frame
	NIC offload	tso on gso on gro on	
環境全体	Storage controller	2つのコントローラに4VMずつ分散	
	Storage性能	不明	問題なし
	Flow control	off	on
	NFS mount	NFS4, rsize/wsize=1048576	
※1 受信拠点側も要調整			

距離の離れた複数拠点の NFV(東京, 北海道)を用いて, 1 台の物理サーバにはそれぞれ KVM を使用し 2 台ずつ VM を構築した. 8K 映像素材は各拠点の共有 NFS サーバに蓄積され, 各 VM が 1 ストリーム(配信性能 3Gbps)を読み出し, 配信する.

サーバアプリケーションは各 VM 上で動作し, 8VM で一斉送信を行うためのクライアントアプリケーションは東京の物理サーバのローカル環境に配置した. また各 VM の映像読出しにおいては物理サーバのリソースが競合しないよう SR-IOV を有効化した. また, カーネルパラメータを表 4 のように決定し, VM の OS を Ubuntu 16.04 から CentOS 6.9 へ, vCPU core 数を 8core から 4core へ, 仮想メモリ保有量を 50GB から 16GB へ変更した. カーネルパラメータは主に NIC のメモリ量の最大値を引き上げ, 伝送における柔軟性を高めた.

表 4 最終構成のカーネルパラメータ

net.core.wmem max = 2147483638
net.core.rmem max = 2147483638
net.core.wmem default = 124928
net.core.rmem default = 124928
net.core.optmem max = 20480
net.ipv4.igmp max memberships = 20
net.ipv4.tcp mem = 1528808 2038144 3057218
net.ipv4.tcp rmem = 4098 1048578 18777218
net.ipv4.tcp wmem = 4098 1048578 18777218
net.ipv4.udp mem = 1535898 2047887 3071798
net.ipv4.udp rmem min = 4098
net.ipv4.udp wmem min = 4098
net.core.netdev max backlog = 4000

#### 4.4 評価実験

これらのチューニングを実施し, SINET NFV 上の 8VM を連携して構築した本システムを用いて, 本学へ向け 24Gbps の 8K 非圧縮映像を伝送する評価実験を実施した. その結果, 図 10 に示すように受信拠点ではノイズのない正常な映像を受信することができた.

また, 従来では揺らぎなく正常な映像が配信出来た時間は数分間であったのに対し, 本研究によるチューニングを実施したシステムは, 数日間に渡って安定的な配信が可能になった.

以上, クラウド上の空きリソースを活用した 8K 非圧縮映像対応の映像配信サーバを実現するため, KVM 上の 8VM を連携してシステムを構築した. 1VM 3Gbps の処理性能を目標に構築し, その安定化のためには SR-IOV の有効化や, VM OS の CentOS 6.9 への変更, vCPU を 4core, メモリの 16GB への削減が有効であることを明らかにした. その結果, 東京と北海道の 8VM を用いて, 24Gbps の非圧縮 8K 映像の安定的な配信が可能となった. 本システムは, NICT 主催の札幌雪まつり 2020 映像伝送実験において展示実験を行い, その実験においても安定動作を確認した.



図 10 チューニング適用後の 8K 非圧縮映像の出力

### 5. 8K 映像ストリームサーバ高度化の提案

ここでは, 2.2 8K 映像ストリームサーバの課題で述べた, 課題 2 と課題 3 の解決方法を提案する.

#### 5.1 提案方式

図 11 に, これらの課題を解決するための提案構成を示す. 本構成では, 映像サーバの直近に IPMR を配置することで, 受信者がいない場合の IP マルチキャストパケットの局所化させることで, 課題 2 の DC 内の帯域の浪費を抑制する. この IPMR は, VNF (Virtual Network Function) として, 物理サーバ上の VM にソフトウェアルータとして構成する. 課題 3 の解決策としては, 他拠点に予備リソースを確保し, 現用拠点のリソースが不足した場合は, 予備リソースで構成した代替映像サーバに映像送信を移行する.

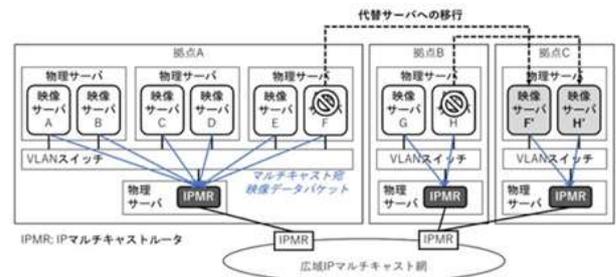


図 11 大容量映像配信システムの高度化の提案構成

#### 5.2 ソフトウェア IP マルチキャストルータの要件

本システムでは映像サーバ1台当たり映像データ 3Gbps, (IP パケットで 3.2Gbps) のトラフィックが発生する. 従って, ソフトウェアルータには, 最低, 映像サーバ 2 台分の 6.4Gbps の IP マルチキャストパケット転送能力が必要となる. また, IP マルチキャスト網への接続のため, PIM-SM と IGMPv2 に対応している必要がある.

#### 5.3 ソフトウェア IPMR の性能検証

これらの要件を満たすソフトウェア IPMR として, NTT コミュニケーションズが開発したソフトウェアルータ Kamuee 及びその商用版の Eenow を実験環境に組み込み, 8K 映像を送受信する検証実験を行った.

図 12 に本実験のネットワーク構成を示す. NFV 内の物

理サーバ2に2つのEenow VMを追加し、新設したVLANで物理サーバ1のVM1とVM2, JGNのIPMRと接続した。VM1, VM2のEenowに接続されたVLANインタフェースから送信された映像データのIPマルチキャストパケットはEenowによりランデブーポイントのJGNのIPMRにルーティングされ、更にKAIT(本学)にルーティングされる。この検証実験において、VM1, VM2から各3.2Gbps(合計6.4Gbps)の映像トラフィックを送信し、NFV内のEenowで6.4Gbpsの転送レートを確認するとともに、大学で受信した映像に乱れないこと確認した。更に、他拠点の8K映像サーバVMからも8K映像パケットを送信し、大学のKamuceで合計25.6Gbpsの転送レートを確認するとともに、大学で受信した映像に乱れないことも確認した。

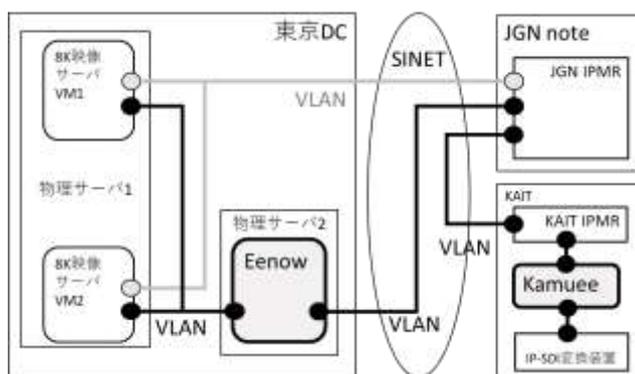


図 12 Eenow/Kamuce の検証実験のネットワーク構成

#### 5.4 代替 VM 切り替え方式

ここでは、8K映像ストリームサーバを構成する8K映像サーバVMの切り替え方式について説明する。

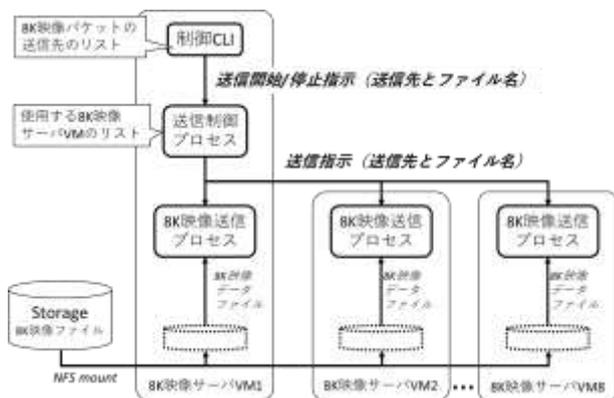


図 13 8K映像ストリームサーバのプロセス構成

8K映像ストリームサーバのプロセス構成を図13に示す。8K映像ストリームサーバは8台の8K映像サーバVMとファイルストレージで構成され、ストレージのファイルシステムは各8K映像サーバVMにマウントされる。全ての8K映像サーバVMには、ファイルシステムから8K映像データを読み出し、宛先に向けて送信、または受信した8K映像データをファイルとして保存する8K映像送受信プロセスがある。8K映像送受信プロセスの制御のユーザインタフェースとして制御CLIがある。送受信制御プロセスは、

使用する8K映像サーバVMのリストを持ち、制御CLIからの送信開始・停止の指示を8K映像送受信プロセスに伝える。従って、送受信制御プロセスの設定ファイルの内容変更により、使用する8K映像サーバVMを切り替えることが出来る。

#### 5.5 代替 VM 切り替え方式と動作検証

2月に開催されたNICT主催の「超高精細映像を用いた広域映像配信実証実験」において、図14に示すように、東京、札幌、大阪の3拠点にEenowを導入し、東京にVM2台、札幌にVM4台、大阪にVM2台の計8台の組み合わせと、東京にVM1台と札幌にVM6台と大阪にVM1台の計8台の2種類の組み合わせで8K映像伝送実験を行った。但し、Eenowの物理サーバの構成上、VM2台分のトラフィックしか収容できなかったため、残りのVMはEenowを経由せず、JGN直結VLANでJGNのIPMRに接続した。

実験期間中、ネットワークトラブルが発生し、全ての検証は出来なかったが、代替VMの切り替え動作は確認することが出来た。

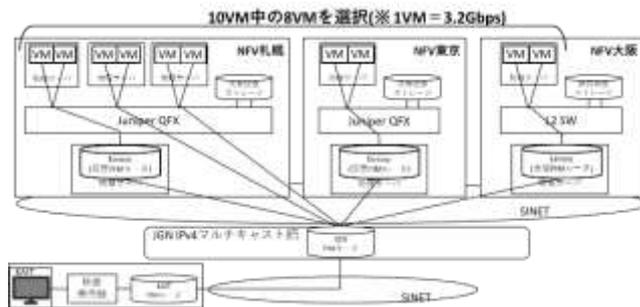


図 14 検証環境の構成

## 6. おわりに

本稿では、(課題1) 18K映像サーバVMの安定動作のための適切なVMリソース配分及びカーネルパラメータのチューニング、(課題2) IPマルチキャストパケットによるDC内の帯域の浪費の抑制、(課題3) 各拠点のリソース使用状況に応じて8K映像サーバVMを配置するシステム化について、解決方法を提案し、検証実験により解決方法の実現性を示した。具体的には、3Gbpsの映像ストリームを扱うVMの安定化のためにはVMのOSにはCentOSを使用し、SR-IOVの有効化や、vCPUを4core、メモリの16GBへの削減が有効であることを明らかにした。その結果、東京、札幌、大阪の計10台の8から、任意の8台を組合せて8K映像ストリームサーバを構成し、24Gbpsの非圧縮8K映像の安定的な配信が可能となった。

また、8K映像ストリームサーバの高度化として、8K映像サーバVMの直近へのIPMRVMの配置による、IPマルチキャストトラフィックの帯域浪費と他ユーザの使用帯域の圧迫の抑制や、リソースが逼迫しているDCからリソースに余裕にあるDCの代替VMを構築し切り替える8K映像サーバVMの再配置の実現性を示した。

OpenStack 等のクラウドコントローラを用いた 8K 映像サーバ VM の IPMR VM のプロビジョニングが今後の課題である。

なお、本実験の一部は、JSPS 科研費 19H04102 の助成を受けて進めた。また、本研究成果の一部は、NICT の委託研究により得られたものである。

## 参考文献

- [1] 青木, 丸山, 瀬林, 石岡, 伊藤, 君山, “広域 SRv6 網を用いた超高精細映像処理のサービスチェイニングの実現” 信学技報, vol. 121, no. 185, pp.1-6, 2021 年 10 月.
- [2] 丸山, 瀬林, 君山, 青木弘太, 小島, 漆谷, 栗本, 河合, 大槻, 小林, “エッジとクラウドの連携による 8K 超高精細映像処理システムの実現”, 信学技報, vol. 121, no. 421, CQ2021-121, pp. 118-123, 2022 年 3 月.
- [3] 池田哲也,丸山充,君山博之,漆谷重雄,栗本崇,河野隆: OpenStack の仮想マシンを用いた 8K 超高精細映像配信サーバの実現, 信学技報, vol.117, no.385, pp.49-54 (2018.3).
- [4] 青木弘太,池田哲也,島仲雄大,瀬林克啓,丸山充,君山博之: 広域分散型 8K 超高精細処理システムにおけるインライン遅延調整機能の実現, 信学技報, vol.118, no.250, pp.123-128 (2018.10)
- [5] 池田, 瀬林, 青木, 丸山, 君山, 栗本, 漆谷, 河野, “KVM 上の仮想マシンを用いた 8K 映像ストリームサーバの実現”, 信学技報, vol. 119, no. 460, NS2019-224, pp. 259-264, 2020 年 3 月.