

# CG映像コンテンツの ライブ型サーバサイドレンダリングに関する検討

## A study on live server-side rendering of CG-based video content

浜口 斉周†  
Narichika Hamaguchi

金子 浩之†  
Hiroyuki Kaneko

道家 守†  
Mamoru Doke

井上 誠喜†  
Seiki Inoue

熊澤 逸夫†  
Itsuo Kumazawa

### 1. はじめに

我々は、テキスト台本をもとに、誰でも簡単に映像コンテンツを制作できる、Text-to-Video という研究を行なっている。TVML(TV program Making Language)スクリプト[1]という映像コンテンツ記述言語を、TVML Player という再生ソフトウェアに読み込ませることによって、リアルタイムCGと音声合成を用いた映像コンテンツを再生・視聴することができる。

さらに、CG映像コンテンツを制作するTVML Playerの機能をWebサービス化[2]することによって、ユーザがソフトウェアをインストールすることなく、インターネット越しにその機能だけを直接利用することができる仕組みを実現した。テキスト台本をWebサービス側に送信すると、リアルタイムCGを描画しながら、描画結果を音声込みのライブストリーミングでユーザ側に返す仕組みである。

しかし、リアルタイムCG描画とライブストリーミングは一般的なテキスト系Webサービスと比較すると、高負荷であるため、負荷分散とスケーラビリティ確保が重要である。また、サーバサイドで描画した結果を返す仕組みであるため、ローカルPCで映像コンテンツ制作の機能を使用する場合と比較して、遅延が発生することは避けられない。

そこで本稿では、開発した映像コンテンツ制作Webサービス(以後、Text-to-Video Webサービスと記す)について、許容できる処理負荷および遅延に関する評価を行なったので報告する。

### 2. Text-to-Video Webサービスのシステム

図1に、開発したText-to-Video Webサービスの概要を示す。Webサービスは、レンダリング管理サーバ、レンダリング+エンコーディングサーバ、ストリーミングサーバおよび素材データベースで構成されており、WebサービスへはRESTインタフェースを通してアクセスする。処理は以下のような流れで行なわれる。

- ①ユーザは制作用WebサイトでTVMLスクリプトを作成し、レンダリング管理サーバに送信する。
- ②レンダリング管理サーバは受信したTVMLスクリプトをバックエンドのレンダリング+エンコーディングサーバに送信するとともに、ストリーミングアドレスを決定して、制作用Webサイトに返す。
- ③レンダリング+エンコーディングサーバではTVMLスクリプトを受け取ると、素材データベースから必要な素材データをダウンロードし、TVML Player用いてリアルタ

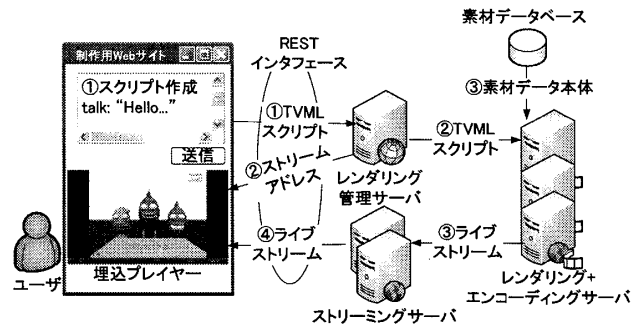


図1 Text-to-Video Webサービスの概要

イムでCG描画・音声合成を行ないつつ、同時並行でライブエンコードを行ない、ライブストリームとしてストリーミングサーバに送信する。

- ④ストリーミングサーバはレンダリング管理サーバに割り当てられたストリーミングアドレスで配信を行ない、制作用Webサイトの埋込済みプレイヤー(Flash Player)で受信・再生する。

### 3. システムの評価

開発したシステムはユーザがソフトウェアの追加インストールをせずとも全ての機能にアクセスできるようにするため、サーバサイドでリアルタイムCG描画・音声合成を行ないつつ、同時並行で描画・出力結果をライブストリーミングでユーザ側に返すという仕組みになっている。従って、サーバサイドには高い負荷が要求される処理であるため、描画リクエストを同時に処理できる本数や、描画リクエストが増えた場合でも負荷分散できるようなスケーラビリティが求められる。また、映像コンテンツ制作機能がローカルコンピュータにある場合と比較して、遅延が発生する。

そこで本章では開発したシステムにおいて、許容できる負荷(描画プロセス数)および、遅延(再生までにかかる時間)に関する評価について述べる。

#### 3.1 許容できる描画プロセス数について

ここでは、サーバハードウェアでどの程度の描画プロセス数を許容できるかを見極めるため、同時描画プロセス数ごとにサーバハードウェアの動作状態をモニタし、負荷の許容度を明らかにするための実験を行なった。実験では、レンダリング+エンコーディングサーバとしてWindows Vista Ultimate(32bit)、Intel Core2 Quad(2.83GHz)、4GBメモリ、nVidia Quadro FX 3700(VRAM: 512MB)という構成のハードウェアを用いた。また、描画サイズは640×360、エンコードはFlash Media Live Encoderを用いてVP6(2Mbps)で行なった。実験に使用したTVMLスクリプトはポリゴン

†日本放送協会 放送技術研究所, Science & Technology Research Laboratories, Japan Broadcasting Corporation

‡東京工業大学, Tokyo Institute of Technology

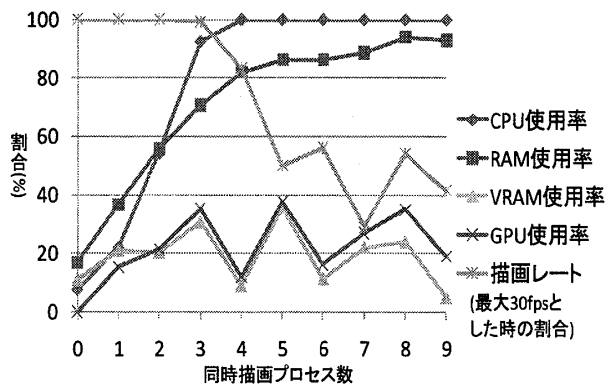


図2 描画プロセス数ごとの動作状態

数・キャラクタ数などの負荷が異なる3種類のものを用いた。モニタリングした項目は、CPU・RAM・VRAM・GPUの各使用率および、CGの描画レート(30fpsを100%とした時の割合)である。

図2に実験結果を示す。結果より、CPU使用率、RAM使用率が上がるにつれて描画レートの低下が起こり、実験したシステム構成では同時描画プロセス数が4本を超えると急速に低下している。RAM使用率が85%を超えるとスワッピングが頻発したため、それが描画レートの低下を引き起こしていると考えられる。

開発したシステムでは、レンダリング+エンコーディングサーバのハードウェアを複数設置し、描画プロセスを振り分ける仕組みであるため、システム全体として処理できる描画プロセス数はハードウェアの数を増やして対応することができる(スケールアウト)。一方で、ハードウェア1台あたりの許容できる描画プロセス数は、実験結果よりCPU・メモリにボトルネックがある可能性が高い。従って、CPU・メモリを効率的に使用するソフトウェア構成にするとともに、ハードウェア性能の増強も必要である(スケールアップ)。今回は32bit OS環境であり、32bit環境では4GBを超えるメモリは使用できないため、メモリの増強には64bit環境に対応する必要がある。

### 3.2 遅延について

次に、映像コンテンツ制作機能をサーバ化したことによる遅延について評価する。実際にText-to-Video Webサービスを用いて映像コンテンツを再生した場合と、ローカル環境で再生した場合とを比較するため、ユーザ側で再生ボ

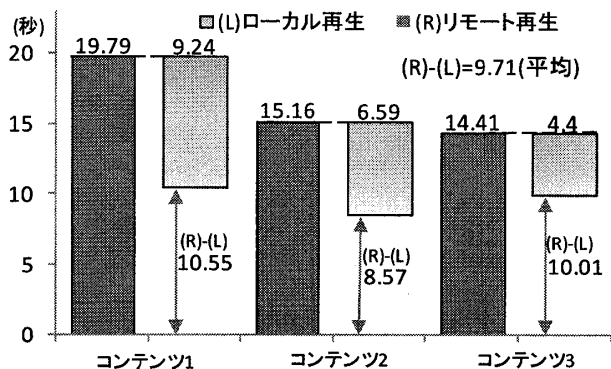


図3 デレイの測定結果

タンを押してから、表示・再生が始まるまでの時間を測定した。ハードウェア構成、コンテンツは3.1と同じものを用い、LAN環境で実験を行なった。

図3に測定した結果を示す。結果より、ローカル再生と比較して再生までの時間が約10秒程度増加することが分かった。サーバログより、10秒の内訳を分析すると、TVMLスクリプト送受信に1秒、エンコーダプロセスの起動に5秒、エンコードに1秒、ライブストリームの送受信に3秒程度かかっていた。

この10秒という遅延は、頻繁に再生・停止を行なうインタラクティブな作業には対応し難いが、テキスト台本を書いてプレビューするような、比較的スタティックな作業には対応できると思われる。

しかし、ローカル再生との比較で10秒、全体で再生までに20秒かかっているコンテンツもあることから、短いとは言い難い。この10秒の遅延を短縮するためには、エンコーダの起動時間と配信による遅延を低減させる必要がある。

今回開発したシステムでは、描画リクエストが来るたびにエンコーダプロセスを起動してからエンコードを行っていたが、エンコーダを常駐させ、描画リクエストを受信してすぐエンコードを開始させれば、エンコーダの起動時間を削減することができる。

また、今回のシステムでは既存のFlash Media Live Encoderをそのまま用いたが、TV電話会議で使われるような低遅延のエンコーディング方式を用いることによって、配信における遅延も短縮できる可能性がある。

### 4. まとめ

本稿では、テキスト台本から映像コンテンツを制作できる、Text-to-Video Webサービスについて、許容できる同時描画プロセス数と遅延という観点で評価を行なった。開発したシステムでは、レンダリング+エンコーディングハードウェア1台あたり4本の描画プロセスを同時に走らせることができ、ハードウェアのスケールアウトやスケールアップによって描画リクエスト数の増加に対応できる。また、ローカル環境と比較して再生開始までの時間に約10秒の増加があり、作業の種類によっては許容範囲内であるが、さらなる遅延の短縮が必要であることが分かった。

本稿では負荷が高くなるレンダリング+エンコーディングサーバに絞って検討を行なったが、今後はシステム全体としての許容できる描画プロセス数、スケーラビリティを評価していく。また、より一層の遅延の短縮を進めていく予定である。

### 参考文献

[1] TVMLウェブサイト <http://www.nhk.or.jp/str/tvml/>  
 [2] 浜口, 金子, 道家, 井上, 熊澤: "RESTful CG映像コンテンツ制作システム," 信学総大, 通信2, B-19-15, pp.576(2010)