

# クラウドを利用した 大規模動画配信システム

菊池正人\* 細田洋佑\*  
岡村 基\* 小池重男\*\*  
池原 勉\*

## Large-scale Movie Streaming System Based on Cloud Service

Masato Kikuchi, Motoi Okamura, Tsutomu Ikehara, Yousuke Hosoda, Shigeo Koike

### 要 旨

三菱電機インフォメーションシステムズ株(MDIS)では、大規模動画配信システムを、アマゾンのクラウド(Amazon Web Service : AWS)を利用することで低コスト、かつ短期間で構築した。

このシステムの特長は、入力した1種類のライブ映像をシステム内で3種類の配信方式に変換することによって、パソコン・携帯電話・スマートフォン向けにライブ映像を提供できることである。また、パソコンライブ映像の最大同時視聴者数を40,000人とする大規模配信を可能としている。

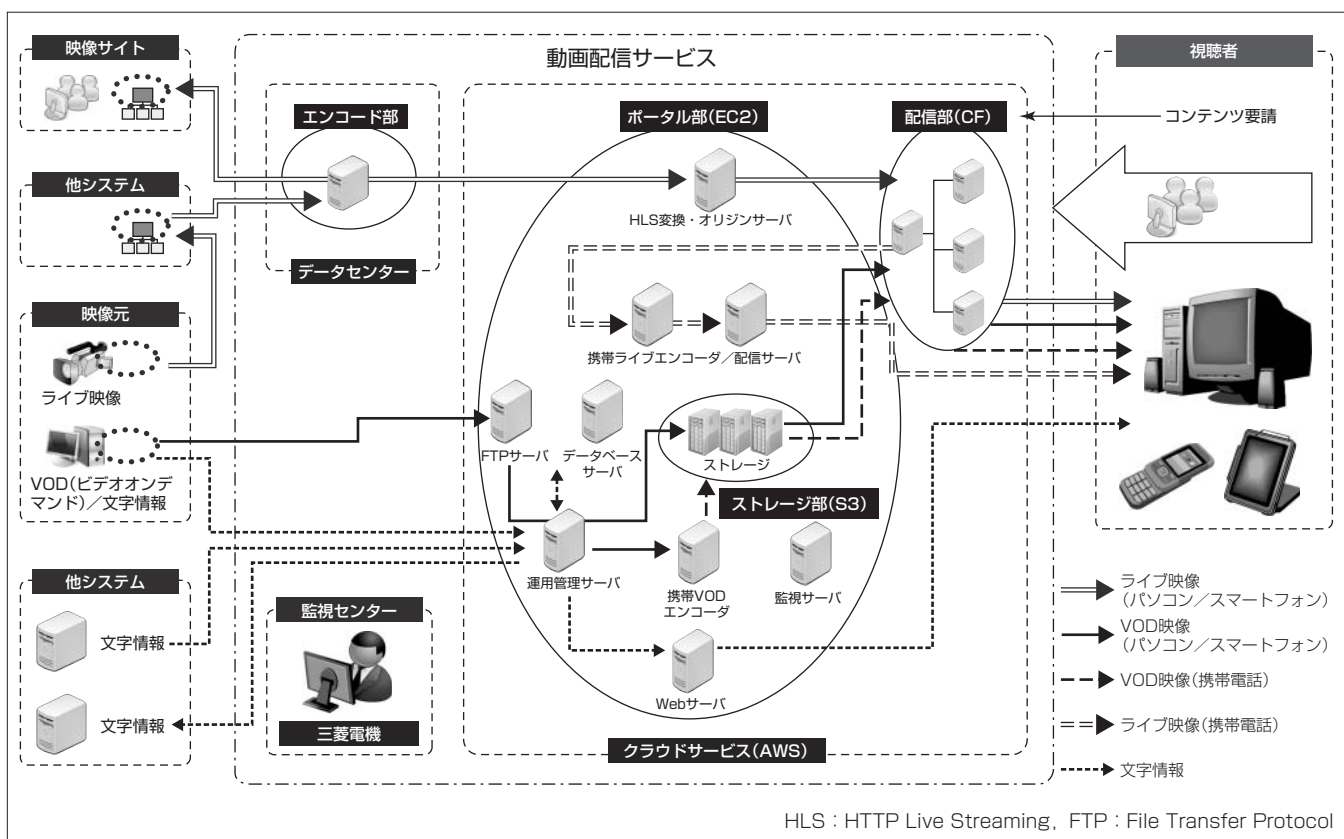
運用面では、クラウド内で持つ監視サービスに加えてオープンソフトウェアであるZabbix<sup>(注1)</sup>を併用することでシステムの各種異常を自動で検知して保守要員等にメールで

通知する監視機能を実現し、従来の有人監視からシステムによる無人監視に切り換えている。

可用性では、クラウドのインスタンス(仮想サーバ)の稼働率が99.95%なので、このシステムの目標稼働率である99.99%を実現するためにクラウド内のデータセンター間で冗長化するシステムを構築している。

また、AWSサービスの特徴である利用時間による課金への対応として、時間帯で自動台数制御を行う方式を採用した。その他、データベースフェールオーバーの遅延、ライブ・オンデマンド映像配信、アクセス数集計等、幾つかの課題に対応している。

(注1) Zabbixは、Zabbix SIA社の登録商標である。



### 大規模動画配信システム

ライブ映像の配信では、外部からライブ映像のデータを受信し、EC2 (Amazon Elastic Compute Cloud) の各インスタンス上で各メディア向けの変換を行う。変換処理を行ったライブ映像データは、AWSのCDN (Contents Delivery Network) サービスであるCF (Amazon CloudFront) を利用して配信する。一方、オンデマンド映像は、S3 (Amazon Simple Storage Service) に一旦格納して、CFで配信する仕組みである。

## 1. ま え が き

MDISでは、2010年11月にオンプレミスと社外CDNサービスを組み合わせた大規模動画配信システムを開発し、ライブ映像・オンデマンド映像をパソコン・携帯電話で一般ユーザーが視聴できるサービスを提供した。それがこのシステムの前身である。

このシステムでは、①システムコスト削減のため、オンプレミスからクラウドへ移行、②近年のユーザー数拡大への対応のため、パソコンライブ映像の同時視聴者数を20,000人から40,000人に拡張、③利用者が増加しているスマートフォン向けのライブ・オンデマンドサービスを追加、の3つを目的としてシステムを短期間で再構築し、2014年8月からサービス提供を開始している。

本稿では、クラウド(AWS)を利用した大規模動画配信システム構築における課題とその解決策について述べる。

## 2. システムの特長

このシステムの特長は、20種類の異なるライブ映像をパソコン・携帯電話・スマートフォン向けに提供できることである。配信方式としてHDS(HTTP(HyperText Transfer Protocol) Dynamic Streaming)とHLS(HTTP Live Streaming)を用意することで、パソコン・スマートフォン(全てのAndroid<sup>(注2)</sup>端末とiOS端末)をサポートしている。また、携帯電話に関しては配信サーバと携帯電話側アプリケーションを一对とした独自方式とした。

一方、オンデマンド映像の配信に関しては、スマートフォンでの視聴を考慮して、前身のシステムが採用したWMV(Windows Media<sup>(注3)</sup> Video)形式からMP4(MPEG(Moving Picture Experts Group)-4)形式に変更した。また、携帯電話への配信は、3つのキャリアブランド(docomo<sup>(注4)</sup>、au<sup>(注5)</sup>、SOFTBANK<sup>(注6)</sup>)に対応するため、パソコン向けのオンデマンド映像から自動的にそれぞれの形式に変換・蓄積することによって、ほとんどの携帯電話で視聴可能となっている。

(注2) Androidは、Google Inc. の登録商標である。  
 (注3) Windows Mediaは、Microsoft Corp. の登録商標である。  
 (注4) docomoは、日本電信電話株の登録商標である。  
 (注5) auは、KDDI株の登録商標である。  
 (注6) SOFTBANKは、ソフトバンク株の登録商標である。

## 3. クラウド化における課題と解決策

この章では、先に述べたシステムの目的、特長を実現するための設計上の留意事項を3.1節で述べ、3.2節から実装上の主要課題と解決策について述べる。

### 3.1 設計上の留意事項

設計時点で特に留意した事項は次の5項目である。

#### (1) 設備投資の最小化

エンコード部での最新機器の採用とクラウドサービスの活用によって、初期設備を通信環境(回線・ルータ)、運

用・監視用パソコン、入力画像エンコード機器のみとした。

#### (2) パッケージ利用とライセンス管理対策

構築スピードとコストの面から、携帯電話サービス対応、映像視聴用画面処理、映像配信・変換処理には既存パッケージをクラウド上で動作検証、チューニングを実施した上で利用した。なお、一部パッケージのライセンス管理で認証に物理メディアが必要であったが、パッケージのカスタマイズによって物理メディアでの認証を不要とし、クラウド上での利用を可能とした。

#### (3) クラウドサービス停止時の対策

クラウドサービス(AWS)全体が停止した場合でも、最小限のライブ映像サービスが継続できるように、オンプレミス環境上のエンコード部から第三者の映像配信サービスへ映像を提供できる機種を選定して、設計した。

#### (4) 映像画質のHD化への対応

現状は縦横比4:3のSD(Standard Definition)画質での映像サービスであるが、近い将来に16:9のHD(High Definition)画質での映像サービス開始が想定されるため、映像入力インタフェース、エンコード機器、Web画面設計などでHD化対応を前提とした設計を行った。クラウド内のサーバリソースについても、HD化による負荷増加に柔軟に対応可能な構成とした。

#### (5) 更なる大規模動画配信への対応

今回構築したシステムでは、配信部からの同時配信人数を最大40,000人としているが、スマートフォンを中心とした同時配信人数の更なる拡張にも耐えられる構成とした。

配信規模拡張への対応として実施した項目は次のとおりである。

##### ① 配信部

現状でも十分な配信容量を保持しているが、バースト的な配信要求急増時に備え、配信部の利用制限値を見直し、今後の定常的な要求増加にも対応可能とした。

##### ② ポータル部

クラウド内サーバリソースについてはスケールアップ(インスタンス増強)、スケールアウト(インスタンス数増加)のどちらの方法でも能力増強が可能な構成とした。

### 3.2 可用性向上への対策

このシステムの目標稼働率は、99.99%であるが、クラウド部の公称インスタンス稼働率は99.95%である。そこで、次の3項目の対策を実施して、動画配信サービス全体としての稼働率99.99%を確保する設計を行った。

#### (1) ポータル画面用Webサーバの耐障害性対策

全体で最大10台の冗長構成を採用した。5台ずつを異なるデータセンター上に分散して配置することで、Webサーバ単位での冗長構成のみではなく、クラウドで利用するデータセンターを含めて耐障害性を向上させた。

#### (2) ライブ配信用オリジンサーバ切換え時の対策

動画配信ソースデータの提供元となるオリジンサーバについても、異なるデータセンター上に分散配置したホットスタンバイ構成として、障害発生などによる切換え時の映像配信タイムラグの短縮を図った。

(3) ポータル部の文字情報連携処理の二重化対策

他システムとの通信を伴うポータル部の文字情報連携処理は、接続先システム側の制約によってプライベートIP (Internet Protocol)での通信が必要であったが、クラウド側の制約もあり、完全な二重化構成での実装ができなかった。対応策として、稼働系と待機系を別のデータセンター上に配置するコールドスタンバイ構成を採用した。

3.3 監視の無人化(自動化)に向けた対策

このシステムでは、24時間無人監視を目的に、自動監視機能を搭載する設計とした。AWSには監視サービス (CloudWatch)が提供されているため、AWS上のサーバインスタンスの稼働状況を監視することでメトリクスを取得することができる。メトリクスとはサーバのリソース値を

意味し、CPU (Central Processing Unit) 使用率、ディスク I/O (Input/Output) 回数、ネットワーク流量、データベースコネクション数などを取得できる。取得されたメトリクスはAWSコンソールに蓄積され、しきい値を超えている場合に警告メールを自動送信する。またAWSコンソールにログインすることによって、これらのメトリクスを、**図1**のようなグラフ上で確認することができる。

このシステムでは、CloudWatchに加え、統合監視ミドルウェアであるZabbixとの連携による監視も行っている。監視システムの構成を**図2**に示す。

CloudWatchとZabbixの連携によって次の3つの機能を実現した。

- (1) CloudWatchとZabbix連携による監視対象の拡大  
CloudWatchでは、メモリ使用率やディスク使用率など、一部取得できないリソース情報があるが、監視対象インスタンスへZabbixエージェントをインストールすることで、これらのリソース情報を監視可能にした。一方、RDS

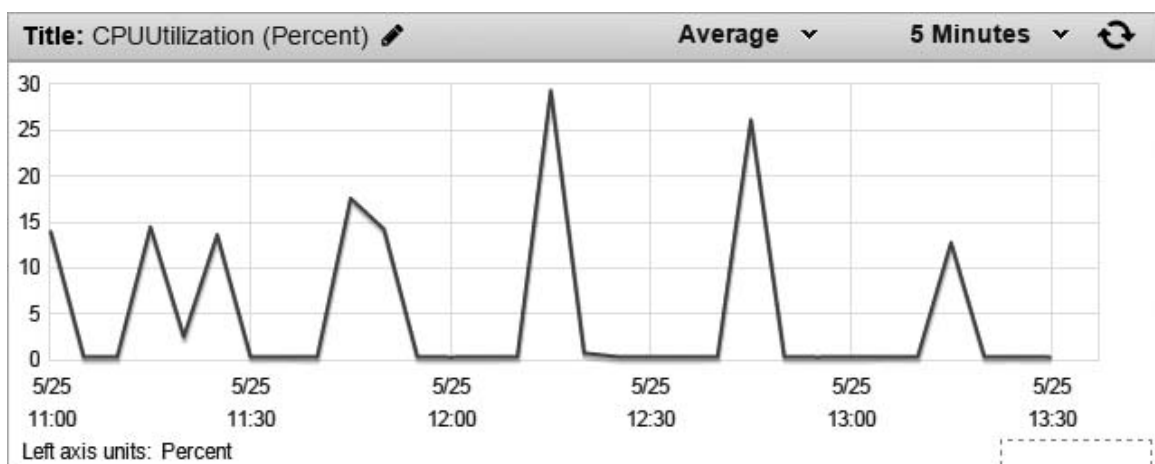


図1. AWSコンソール上でのメトリクスグラフ(CPU使用率)

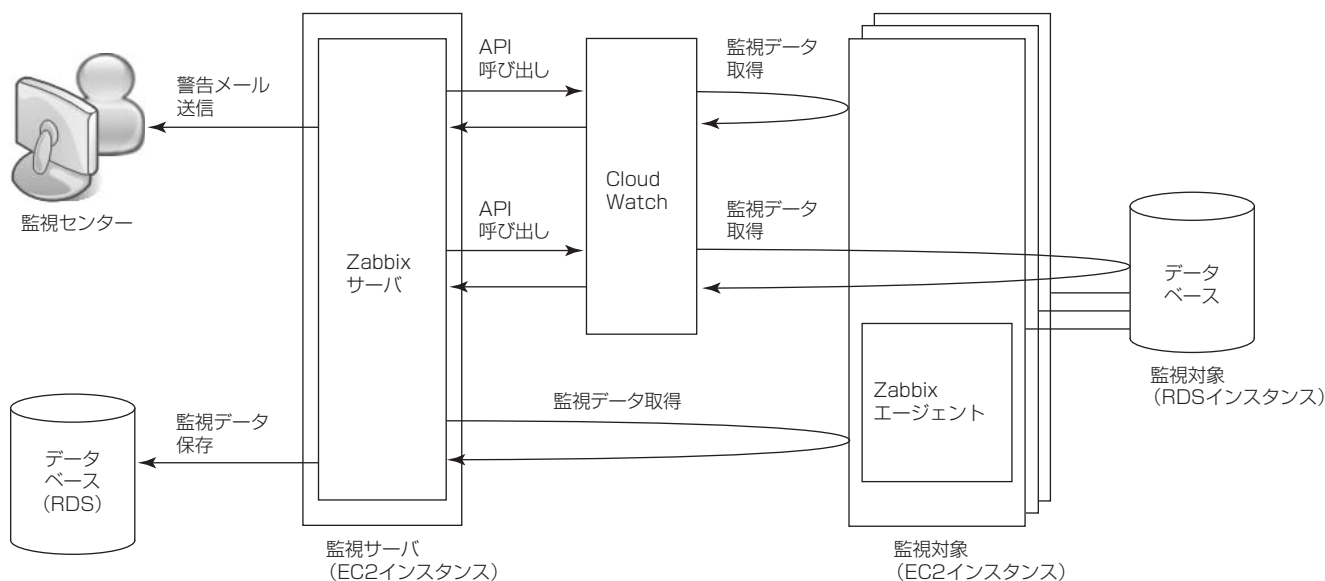


図2. 監視システムの構成

(Amazon Relational Database Service for Oracle Database)など、OSへのログインができず、Zabbixエージェントを導入できないインスタンスについては、Zabbixエージェントでリソース情報を取得することができないため、CloudWatch経由で情報を取得することで対応を図った。このように、CloudWatchとZabbixを連携させることで、両者単独では監視できない、保守サポートに必要な監視データを取得できるようにした。実際に監視している項目の一部を表1に示す。

(2) メトリクス保存期間の拡大

CloudWatchで取得したメトリクスをAWS上で保存する場合、最長2週間までしか保存されない。そのため、このシステムではCloudWatchで取得したメトリクス、及び、Zabbixエージェントによって取得した監視データをZabbixサーバに接続されたRDSインスタンスに一元的に保存する設計とした。その結果、最長5年間保存できるようになり、稼働状況調査や障害調査時に、過去の監視データをさかのぼって参照できるようになった。

(3) 障害検出時のインスタンスの自動制御

Zabbixでは外部チェックと呼ばれる機能によって、監視データがしきい値を超過した場合等をトリガー条件に設定して、外部コマンドを実行することができる。また、AWSにはAWS CLI(Command Line Interface)と呼ばれるコマンドラインベースのAPI(Application Programming Interface)が提供されており、APIの呼出しによってインスタンスの制御が可能である。このシステムではZabbixの外部チェック機能を用いてAWS CLIを呼び出すスクリプトを実行するようにした。これによって表2に示すようなインスタンスの自動制御が可能となった。

3.4 データベースフェールオーバー遅延対策

このシステムでは、外部システムから送信される文字情報をその都度データベースに格納し、1分周期で表示情報を更新して提供する必要があった。

このシステムでのデータベースは、AWSのRDSサービスを信頼性向上のためMulti-AZ(Availability Zone)配置オプションを利用して構築したが、このRDSの冗長化構成ではフェールオーバー時にDNS(Domain Name System)への付け替え等が発生する関係で、フェールオーバーに数分かかることがあり、情報更新が滞るという課題があった。

そこで、Webサーバ側の処理を見直して、データベースから情報取得して表示データを生成する部分については、最後に受け取った情報を再利用して情報表示を継続する仕組みにすることで対応した。

3.5 インスタンスの稼働制御によるAWS課金対策

AWSではインスタンスを利用した時間に応じて課金される。そのため、システムの負荷に関係なく常に全てのインスタンスを稼働させてしまうと、無駄なランニングコス

表1. 監視項目

監視項目	CloudWatch単独での監視	Zabbix単独での監視
CPU使用率	○ <sup>(注7)</sup>	○ <sup>(注7)</sup>
メモリ使用率	×	○
ディスク使用率	×	○
RDS監視	○	×
ELB監視	○	×
ログ監視	×	○

○：監視可 ×：監視不可

ELB：Elastic Load Balancing

(注7) CPUの全体使用率は、CloudWatch、Zabbixともに監視可であるが、CPUの詳細情報(user, system, iowait, load average等)に関しては、CloudWatchは監視不可、Zabbixは監視可である。

表2. Zabbixによるインスタンス制御(一部の例)

項目	トリガ条件	制御方法
インスタンスの再起動	Zabbixサーバからインスタンスへping疎通に失敗した場合	Zabbixサーバから該当インスタンスに停止コマンドを実行して、インスタンスを再起動させる
待機系インスタンスへのフェールオーバー	ZabbixサーバからAWSインスタンスへのweb監視で異常を検知した場合	ZabbixサーバからRoute53(DNS)のCNAMEレコードを待機系インスタンスのホスト名に書き換えるコマンドを実行して、AWSへのアクセスを、現用系から待機系へ切り換える

トがかかるという課題があった。必要な時に必要な数だけインスタンスを使用するためのAWSの標準機能として、自動スケーリング機能がある。これは、インスタンスの負荷があらかじめ設定した基準に達した場合に、EC2インスタンスの負荷に合わせてインスタンスのスペックを上下させ、インスタンス数を増減させるものであるが、このスケーリングは起動開始から起動完了となるまでに数分を要していた。

また、このシステムには、1日を通して定期的に視聴者数が急増して負荷が高くなる時間帯があり、その時間帯に限りインスタンス数を増やしたいが、急激な負荷増大には、自動スケーリング機能のスピードが追い付けないという課題があった。その対策として、AWSのAPIにインスタンスを起動/停止できるものがあり、これを用いてインスタンス数を増減させる方式を採用した。

1日24時間を3パターンに分け、日中を全てのインスタンス稼働、夜間を半数のインスタンス稼働、深夜を必要最低限のインスタンス稼働とした。それによって、常に全てのインスタンスをフル稼働させることがなくなり、ランニングコストを削減し、AWSの無駄な課金が生じないようにした。

3.6 アクセス数集計の精度向上への対策

このシステムでは同時視聴者数の把握として、視聴されたアクセス数から同一IPアドレスでアクセスされた場合と同一チャンネルをアクセスした場合は、1回とカウントする仕様で5分ごとに統計情報を収集する仕組みが要求されていた。

また、ライブ映像に関しては、外部システムが直接配信ポイントを参照してライブ視聴する仕組みを提供するが、システム内でのアクセス数をカウントする方法では、その

表 3. ライブ映像サービス提供対象と配信形式

サービス対象	配信形式	備考
パソコン, 一部スマートフォン	HDS	Android2.2以前
スマートフォン	HLS	
携帯電話	独自形式	docomo機種のみ

ような直接参照されるアクセス数が除外されてしまうという課題があった。この課題に対しては、AWSが提供するサービスの1つである、配信部(CF)の各サーバのアクセスログをS3に収集する機能を利用して、それらのアクセスログを解析することによって統計情報を算出する仕組みを構築することで対応を図った。

また、このログ収集は、AWSの仕様では60分程度で配信部の各サーバからS3に格納されるとされていたが、実際には、24時間以降に格納されるサーバがあることが判明したため、このシステムでは、72時間後にログ解析を行う仕組みとしている。

3.7 ライブ・オンデマンド映像の多様な配信形式への対応

ここでは、多彩な機種で動画視聴を可能とするために行った設計上の留意事項について述べる。

3.7.1 ライブ配信形式

このシステムはSD-SDI(Standard Definition-Serial Digital Interface)で受け取った映像・音声を最終的に表3に示す形式で視聴者へサービス提供している。

HDSでの提供先は規格の定義がなく、代表的なAndroid搭載機器での検証を実施し、ユーザーエージェントによる対象機器の判定と配信形式の選択を実施している。

携帯電話は、先に述べたとおり、クラウド上での動作確

表 4. オンデマンド映像サービス提供対象と配信形式

サービス対象	配信形式	備考
パソコン, 一部スマートフォン	MP4ファイル	
携帯電話	独自形式(7種類)	docomo, au, SOFTBANK

認及び入力映像対応カスタマイズを経てライブ配信パッケージをdocomo製品のみに対応させた。

なお、パソコン、携帯電話については配信形式に合わせた再生プレーヤーアプリケーションの更新も実施している。

主にスマートフォン向けに配信しているHLSについては、クラウド内インスタンスにHDSからの変換処理を実装しており、変換が不要なHDSに比べて即時性が損なわれる面があるが、変換処理に利用するインスタンスのチューニング及び無線環境での利用が大多数であることなどから、パソコンライブとの差異は目立つものとはなっていない。

3.7.2 オンデマンド配信形式

ライブ配信に対して、オンデマンド配信では元映像はライブ映像からの取得ではなく、ダイジェスト版への編集後にMP4ファイルとしてこのシステムへ登録後、配信する方式としている。基本はMP4ファイルをそのまま配信しているが、携帯電話のみ表4に示すとおり、機種個別対応の変換実施後に配信している。

4. む す び

クラウドサービスを利用した大規模動画配信システムの構築は、ビジネスシーンで要求が高まってきている分野である。今回の事例・構築ノウハウをいかして今後のシステム構築に取り組んでいく。