

特集号
招待論文

サービス制御システム(IMS)のクラウド化とテストベッド検証

—IMSの省電力化と安定稼働—

伊藤 学^{†1} 小森田 賢史^{†1} 北辻 佳憲^{†1} 横田 英俊^{†1} 林 偉夫^{†2}^{†1}(株) KDDI研究所 ^{†2} 日本電気(株)

移動体通信システムへクラウドの概念を導入することができれば、稼働中のシステムであっても柔軟に構成を変更することで消費電力を削減できるほか、障害発生時においても代替サーバでサービスを引き継ぐことができる。本論文では、このような省電力性と高信頼性を兼ね備えた移動体通信システムを実現するために、サービス制御システム(IMS: IP Multimedia Subsystem)を構成するサーバ間でのセッション情報の移動方式を提案する。提案方式によってシステムを動的に構成変更できることをテストベッドを用いて実証し、同方式の実用性について論じる。

1. はじめに

移動体通信事業者は、電話・インスタントメッセージ・ライブ動画共有・コンテンツ交換等のサービスをIP上で品質を保証して提供するために、IPベースのサービス制御システム(IMS: IP Multimedia Subsystem) [1]の導入を進めている。従来の電話サービスに加えて、上述の多様なサービスを提供する場合、IMSにて処理すべき信号量の増大が予測されるが、最繁時の処理に必要な設備容量を見積もることは容易ではない。予測困難な信号処理(呼処理)の負荷変動に対して、サービスごとにサーバ設備を配備するような従来の移動体通信のシステム構築では、過剰な設備投資を抑えることができない。筆者らは、必要な処理能力に応じて最適な設備容量の確保を実現するクラウドの概念を移動体通信システムへも導入することが望ましいと考えている。

IMSを構成するサーバ群を仮想化し、物理サーバの動的な割り当てができれば、呼処理の負荷変動に応じて物理サーバの稼働台数を最適化することが可能となる。また、管理するサービス処理状態(セッション情報)の移動を仮想サーバ間で柔軟に行うことができれば、よりきめ細かい最適化により稼働サーバの台数を最少化でき、システムの省電力化が可能となる。さらに、物理サーバの障害に際してセッション情報を他の物理サーバへ引き継ぐことができれば、安定したサービス提供が可能となる。

本論文では、セッション情報の移動方式を提案し、同方式によりIMSを構成するサーバの動的な構成変更方式

と、障害に際して多量のセッション情報の復元を可能とする実装方法を示し、これらをテストベッド上で実証する。これにより、従来システムよりも消費電力の削減と信頼性の向上を実現するサービス制御システムの実用性を論じる。セッション情報の移動方式については、サーバのセッション情報配置と通信の経路制御(OpenFlow[2])を連携することで、標準の呼処理手順を逸脱することなくセッション情報の移動を実現する。

以降、第2章ではIMSの概要と構成変更の機能要件を述べ、第3章でセッション情報移動のためのOpenFlowによるサービス制御メッセージ経路制御方式を論じる。次に、第4章で多数のセッション情報の移動を可能とする実装について、第5章でテストベッドによる実証実験について述べる。最後に、第6章でまとめとする。

2. IMSのクラウド化の機能要件

2.1 IMSとセッション制御

図1にIMSの概要を示す。IMSは、SIP(Session Initiation Protocol) [3]を用いてサービス制御を行うCSCF(Call Session Control Function)、加入者情報を管理するデータベースのHSS(Home Subscriber Server)、他の移動体通信事業者との接続点であるIBCF(Interconnection Border Control Function)、セキュリティ機能(アクセス制御、トポロジ隠蔽)やプロトコル差分吸収、メディア変換・監視等を提供するSBC(Session Border Control)で構成される。CSCFは、主体となってサービスの制御

と状態管理を行うS-CSCF (Serving CSCF), 加入者端末であるUE (User Equipment) と直接通信を行うP-CSCF (Proxy CSCF), (電話の場合に) 着呼UEを収容しているS-CSCFへSIPメッセージの転送を行うI-CSCF (Interrogating CSCF) に大別される。

UEはREGISTERメッセージ(図1の一重線矢印)を送信することでS-CSCFへ登録を行う。この際、S-CSCFはHSSが保持している加入者情報を用いて認証を行う。登録が成功すると、UEの登録情報がP-CSCFとS-CSCFに保持され、以後UEにより送受信されるSIPメッセージ(e.g., INVITEメッセージとその応答メッセージ)は、これらのCSCFを経由して処理される。図1では、UE1がP/I-CSCF1を経由してS-CSCF1へ登録される状態を示している。電話サービスの発着信を例にするとUE1(発呼UE)とUE2(着呼UE)の間で送受信されるSIPメッセージ(図1の二重線矢印)は、UE1—P-CSCF1—S-CSCF1—I-CSCF2—S-CSCF2—P-CSCF2—UE2の経路で配送される。各CSCFでは、SIPメッセージを基にサービスの処理状態の管理と制御を行う。

本論文では、IMSのクラウド化を、稼働中のIMSを構成する3種類のCSCFの台数をサービス要求に応じて最適化することと定義する。また、サービスの処理状態をセッション情報と呼び、明示的に区別しない限り電話サービスを単に「サービス」と呼ぶ。

2.2 IMSのクラウド化の機能要件

IMSでは、起動したUEは自らをIMSへ登録し、その後は当該UEと登録先P/S-CSCFが対応付けられる。そのため、過負荷になったCSCFに対してUEとの対応付けを変更できないと、同種のCSCFが余剰配備されていたとしてもサービスの継続が困難になる。この対応付けの変更ができれば、複数のCSCFで処理負荷が減少した際

には、セッション情報を同じIMS内に存在する少数の同種CSCFへ集約してCSCFを動作させる基底電力を節約できる。

セッション情報の移動を行う方式として、仮想マシン(VM)のライブマイグレーションを用いる方式[4],[5]が提案されている。これらの方式では、CSCFが動作しているVMを少数の物理マシンへ集約することで、セッション情報の移動を実現している。しかし、VMで多量のセッション情報の処理を行う状況では、ライブマイグレーションの際に大量のメモリ書き換えが発生するためにVMの移動に時間を要し、サービスの中断を引き起こす。一方で少数のセッション情報を処理するVMを多数用いてこの問題に対応する場合には、VMを動作させる消費電力が増大する。よって、IMSの省電力化を実現するためには、多数のセッション情報を収容するCSCFに対して、IMS内に存在する同種のCSCF間で、一つひとつのセッション情報を移動させる機能が必要である(以下では、同種のCSCF間のセッション情報の移動を単に「セッション情報移動」もしくは「セッション情報の移動」と記述する)。

以下に、このようなセッション情報移動に必要な3つの機能要件を説明する。1つのセッション情報の移動に着目すると、セッション情報の移動先CSCFに対向するUEや他のCSCFが、SIPメッセージを適切に転送できる必要がある。このためには、SIPヘッダ(RouteヘッダやViaヘッダ等)内の情報(宛先IPアドレス等)を変更する必要がある。しかし、ローミングする(他事業者から提供される)UEは標準準拠であり、セッション情報移動に伴うSIPヘッダ内の情報変更機能を実装していないために、セッション情報の移動先CSCFへSIPメッセージを適切に転送できない。CSCF間でセッション情報を移動する場合には、SIPメッセージのヘッダ情報を変更

することなくその経路を変更できる機能が必要である(機能要件①)本要件は、経路制御において一つひとつのセッション情報(UE)を区別するために、SIPメッセージ内部のUEを区別する情報を用いる必要であり、移動体通信システムに特有の機能要件である。

また、障害等により停止したCSCFのセッション情報の処理を他のCSCFへ引き継ぐためには、CSCFが稼働中にセッション情報をバックアップし、障害検知後に他のCSCFへセッション情報を直ちに復元(リストア)する機能が必要である(機能要件②)。移動体通信

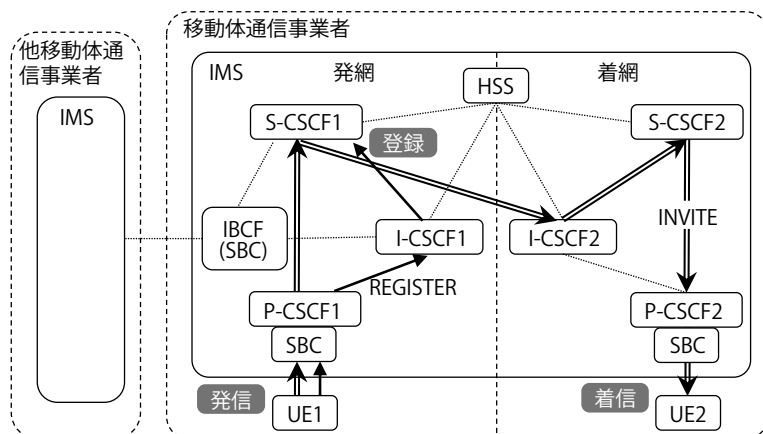


図1 IMSアーキテクチャ

システムの特徴として、CSCFが扱う多数のセッション情報はその処理状態が頻繁に更新される。処理状態の整合を確保してサービス中断を生じさせずにセッション情報をリストアできることが本機能要件には必要である。

さらに、省電力化を目的に稼働CSCFを最少化するためには、稼働中のCSCFの収容状況に応じて、どこのCSCFからどこのCSCFへセッション情報を移動すべきかを決定するセッション情報配置計算機能が必要である（機能要件③）。その方式として、PetrucchiらはVM環境において稼働サーバ数を最少化する構成の導出を行い、不必要なサーバを停止させる方式[6]を提案しているが、VMの移動コスト（消費電力）や移動許容時間が考慮されていない。

第3章に機能要件①～③を実現する方式を説明する。

3. セッション情報移動のための SIP メッセージ経路制御方式

提案方式では、同じ種類のCSCF（P/I/S-CSCF）に同一IPアドレスを割り当て、セッション情報を移動する際にSIPヘッダ書き換えを生じさせないようにすることで標準規定を満たすようにする。この方針を実現するために、ネットワーク装置（スイッチ、ルータ等）ではIPアドレスとは別の識別子を用いてSIPメッセージを転送することが求められる。この識別子としてSIP URI（UE固有の識別子）を用いる。

一方で、昨今のネットワーク装置のデータ転送はEthernetもしくはIPヘッダを用いることが主流となっている。UDPメッセージのペイロードに記述されるSIP URIを読み取るデータ転送は転送性能の大幅な低下を招くため、一つひとつのSIP URIに対応する識別子を用意し、これをEthernetおよびIPヘッダの宛先アドレスとして用いる。

最近では、データの転送部と経路制御部を分離した

OpenFlowの出現により、データの転送経路の動的変更が可能となっている。そこで、移動させるセッション情報の選択とSIPメッセージの転送経路の決定をOpenFlowの経路制御部と位置づけ、また決定経路の指示に基づくSIPメッセージの転送をOpenFlowのデータ転送部に対応付ける。

3.1 セッション情報移動アーキテクチャ

図2にOpenFlowを用いるアーキテクチャを示す。OpenFlowスイッチにてSIPメッセージをUEごとにフローとして識別できるように、SBCの機能を拡張し（図2のeSBC: enhanced SBC）、UEとOpenFlowスイッチの間に配置する。eSBCはUEから送信されたSIPメッセージを受信すると、From/Toヘッダ（REGISTERメッセージの場合はFromヘッダのみ）に含まれるUEのSIP URIから対応する固定ビット列（以下、「ラベル」と呼ぶ）をEthernetのMACアドレスフィールドに書き込む（ラベルの定義は次節に説明する）。このとき、発呼UEのSIP URIに対応するラベルを宛先MACアドレスフィールドへ、着呼UEのラベルを送信元MACアドレスフィールドへ書き込む（REGISTERメッセージの場合は宛

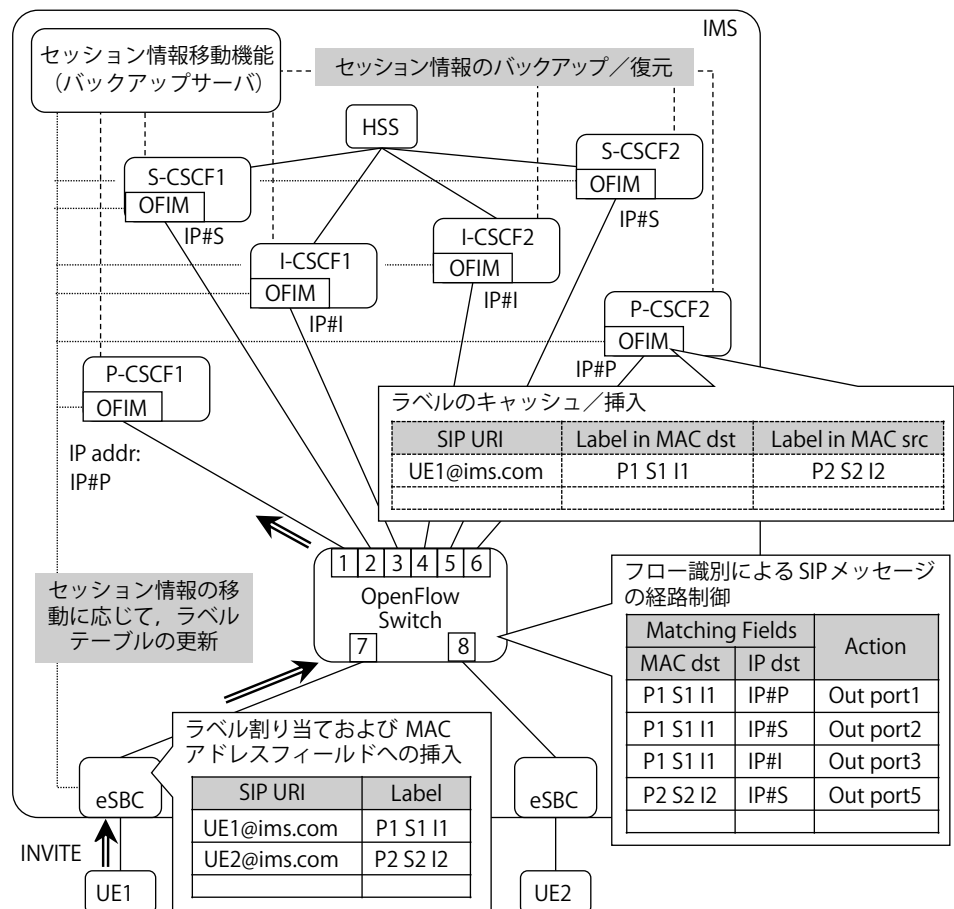


図2 提案アーキテクチャ

先MACアドレスフィールドのみへラベルを書き込む)。
OpenFlowスイッチは、宛先MACアドレス（ラベル）と
宛先IPアドレス（P/S/I-CSCFの種類によってアドレスが
異なる）によってフローとして識別したSIPメッセージ
を各CSCFへ転送する。

CSCFにはSIPメッセージを送受信する際にラベルを維持できるよう新規にI/Fモジュール（OFIM: Open Flow Interface Module）を導入する。このモジュールは、受信したメッセージに含まれるラベルとSIP URIの組をキャッシュする。SIPメッセージを送信する際にはキャッシュしたラベルをMACアドレスフィールドに書き込む。

セッション情報移動機能はバックアップサーバとしても動作し、セッション情報のコピーを保持し、CSCFに障害が発生した際には保持しているコピーを他のCSCFへリストアする。このときOpenFlowプロトコルを用いて、CSCFのOFIMとeSBCのラベルを更新する。

さらにセッション情報移動機能では、IMS全体の省電力化のために、セッション情報移動とCSCFの起動・停止に関する消費電力モデルを用いた線形計画法により、セッション情報移動とCSCFの稼働台数変更を逐次的に行う。しかし、このような構成変更を頻繁に繰り返すと、CSCFの台数を削減できたとしても、セッション情報の移動ならびにCSCFの起動・停止により全体の消費電力が上昇してしまう。そのため、構成変更の過剰な実行を防ぐ周期が重要となる。この周期に関するプラクティスを以下に述べる。構成変更開始からある時刻までの間(周期)の消費電力が、直前の周期の間の消費電力よりも小さくなる場合に、構成変更を実行する。また、処理しているセッション情報数の変動により、CSCFに収容可能なセッション情報数の上限を超えてしまう場合にも構成変更を実行する。構成変更の実行頻度は発呼レートに依存することから、より効果的に省電力化を実現するためには、発呼レートなど実際の環境に応じた周期を設定できることが必要である。

3.2 ラベル割り当て方式

MACアドレスフィールドに記述するラベルはIMSが収容するUEの数（たとえば3,000万）に達するが、OpenFlowスイッチにこれらのラベルを登録することは困難である。そこでラベルを、UEが登録時に割り当てられるP/I/S-CSCFの識別子（ID）の組合せで構成し、このIDを、IMSの拠点識別子と拠点内でCSCFを区別する識別子の組で構成する（図3）。拠点識別子を用いることで、OpenFlow

01	2
N	

図3

スイッチに登録するラベル数を $(S_{P+}N-1) \cdot (S_{I+}N-1) \cdot (S_{S+}N-1)$ にすることができる。なお、ここでは N を拠点数、 S_P , S_I , S_S を拠点内 P/I/S-CSCF の最大の数とする。たとえば、現在の CSCF 製品は 1 台当たり 30 万程度の UE を収容することが可能なため、3,000 万の UE を想定した場合、各 CSCF は 100 台必要となる。各 CSCF を 5 拠点に 20 台ずつ分散して配備する場合には、OpenFlow スイッチが保持するラベル数は $13,824 (= (20+5-1)^3)$ にでき、UE の数 (3,000 万) には依存しない。

このようなラベル数の削減が可能になる理由は、OpenFlow プロトコルならびにスイッチがマスク化したラベルを扱うことができるためである。IMSを複数拠点に分散配備する場合には、SIP メッセージの転送に OpenFlow スイッチを分散し、OpenFlow スイッチは他拠点の内部の CSCF を区別する必要はなく、他拠点の CSCF のラベルは1つの拠点識別子に集約可能である。

4. セッション情報移動方式の実装

本章では、CSCFの突然の障害（無応答の停止等）、ならびにCSCFを複数拠点に分散配備した際の拠点の障害（大規模な通信障害等）に対して、サービス提供を継続する高性能化の実装について論じる。

4.1 セッション情報のバックアップ方式

CSCFの障害に際してサービスを継続させるためには、障害直前のセッション情報をリストアできることが求められる。そのためには、CSCFがSIPメッセージを受信しセッション情報を更新するたびに、そのバックアップが求められる。また、拠点の障害に対応するためには、セッション情報を異なる拠点にバックアップすることが必要不可欠であるが、頻繁なバックアップは通信遅延の累積を引き起こし、CSCFの処理性能を低下させる。

たとえば電話サービスにおける発呼処理ではセッション情報が約60回更新される（詳細は割愛する）が、筆者らはこれを23回に削減可能なバックアップ方式を考案し[7]、本論文で実装を行った。SIPでは、処理要求メッセージに対する応答メッセージが得られない場合に、処理要求メッセージを再送する。多くの運用形態

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
Not used										跳点 ID	P-CSCF ID										跳点 ID	I-CSCF ID										跳点 ID	S-CSCF ID																										

図3 MACアドレスフィールドに割り当てる値

では、500 ミリ秒後に最初の再送をした後は、再送間隔が順次2倍の間隔に拡大される。この再送の特徴を活用すれば、処理要求メッセージにより更新されたセッション情報はバックアップせずに、応答メッセージにより更新されたセッション情報をバックアップすればよい。これは、処理要求メッセージによる更新内容はバックアップせずとも、処理要求メッセージの再送を受信することにより復元できるからである。ここで、CSCF（旧CSCF）が処理要求を処理後に停止し、セッション情報を別のCSCF（新CSCF）でリストアする場合、SIPメッセージ

の特徴である再送の仕組みを考慮し次のような動作で障害直前のセッション情報へ復元されることになる。

1) 処理要求メッセージを送信したUEが応答メッセージを受信できないため処理要求メッセージの再送を開始する。2) 何度かの再送の途中に、新CSCFに処理要求メッセージにて処理される前のセッション情報がリストアされる。3) 新CSCFに何度目かの再送処理要求メッセージが届き、障害直前のセッション情報に復元される。2) において、処理可能になるまでの新CSCFに、再送処理要求メッセージが到着する場合には、新CSCFはそのメッセージを廃棄する。

4.2 eSBC 高性能化のための実装方式

拠点障害が発生した場合、セッション情報のリストアには高い性能が求められる。セッション情報の移動を実現するアーキテクチャ（図2）では、eSBCがボトルネックになる懸念があるため、SIPメッセージ経路のラベル書き換えを高速に行う実装を行った。以下ではその実装について論じる。

eSBCは、SIP URI に対してラベルを割り当て、Ethernetの送信元／宛先MACアドレスに書き込む処理を行う。そのため、REGISTERのSIPメッセージからSIP URIを抽出してラベルを割り当て、以降のSIPメッセージに対しても同じラベルを割り当てる必要がある。このようなSIPメッセージ処理においては、パケットのペイロードからSIP URIを抽出する処理と、SIP URIとラベルの組合せを記憶するセッション管理テーブルの登録・検索処理が性能ボトルネックとなる。

eSBCの高速処理を実現するために、SIPメッセージ処理をCavium社のネットワークプロセッサである

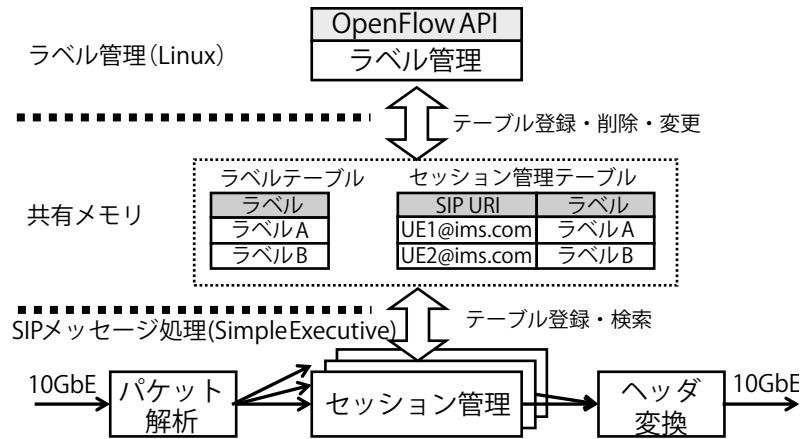


図4 eSBCの構成

Octeon Plus (CN5650^{☆1}) 上に実装した。Octeon Plusは、750MHzで動作するMIPSコアを12個有している（図4）。低オーバーヘッドなOS（Simple Executive）にSIPメッセージ処理を、LinuxにeSBCのラベル管理機能を割り当てた。

SIPメッセージ処理は、受信したパケットヘッダを解析してSIPパケットを判定するパケット解析、SIPメッセージのペイロードに格納されているFrom/Toヘッダを検索して、SIP URIを抽出しラベル割り当てを行うセッション管理、割り当てられたラベルをEthernetの送信元／宛先MACアドレスへ格納するヘッダ変換で構成した。それぞれの処理はCPUコアに割り当て、機能ブロック間はCPU内部バスにより接続される。付与するラベルを管理するラベルテーブルとSIP URIおよびラベルを管理するセッションテーブルは、共有メモリ上に配備し、Linux上で動作するラベル管理機能とSIPメッセージ処理からアクセス可能である。これらのテーブルは、ラベル管理が提供するOpenFlowAPIを用いて、OpenFlowコントローラから制御可能にし、eSBCとOpenFlowスイッチの統合制御を可能とした。

先に述べたようにセッション管理が性能ボトルネックとなるため、以下に示す実装により高速処理を実現した。

(1) SIP URI抽出処理の高速化

From/Toヘッダを高速に検索するための文字列検索アルゴリズムにボイヤムア法（QuickSearch）を採用し、CPUコアあたり10,000ppsの性能を達成した。

(2) セッションテーブルの高速化

セッションテーブルは、SIP URIのハッシュ値をアドレスとして参照できるよう構成し、ハッシュ値のコリジョンが発生した場合は、リンクリストを構

☆1 Octeon Plus プロセッサ, http://www.cavium.com/OCTEON-Plus_CN56XX.html

成する。セッションテーブルは、100万エントリを管理でき、約200,000アクセス／秒の性能を達成した。1パケットあたり1回（検索または登録）のアクセスが発生することを考慮すると、約200,000ppsの性能となる。

(3) 負荷分散による高速化

セッション管理の負荷分散方式は、SIPセッション単位にCPUコアに分散しセッションテーブルを個別に持つ方式と、パケット単位にCPUコアに分散しセッションテーブルを共通で持つ方式が考えられる。SIPセッション単位に負荷分散を行う場合は、SIP URI抽出処理を行う必要があるため、(1) および (2) の結果より、パケット単位に分散する方式がより性能を向上できると考えられる。以上の考察から、セッション管理の負荷分散方式は、パケット単位に処理を分散しセッションテーブルを共通に持つ方式とした。

4.3 eSBC 高性能化に関するプラクティス

eSBCの高性能化を実現するために、ネットワークプロセッサで動作するSimple Executive上にeSBCの機能を実装した。以下にネットワークプロセッサを用いた高性能化に関するプラクティスを述べる。まず、eSBCを機能ブロックに分割し、各機能ブロックをCPUコアへ割り当てeSBCを構成した。機能ブロックの1つであるSIPメッセージ処理機能はオープンソースソフトウェアを採用したが、Makefileの修正のみで移植が可能であった。次に、各CPUコアに割り当てた機能ブロックの処理時間を計測したところ、SIPセッション管理がボトルネックであることが判明したため、4.2節で述べたように負荷分散などの対策で性能ボトルネックを解消した。ネットワークプロセッサは、1コアあたりの処理性能はそれほど高くなく、高性能化を達成するためには複数のCPUコアを利用することが必須となる。そのため、実装する機能をプロセッサの処理能力に合った規模の機能ブロックに分割し構成することが重要となる。また、オープンソースソフトウェアを活用する場合には、Simple ExecutiveなどのネットワークプロセッサのOSにおいて、必要となるAPIが提供されているか事前に把握しておくことが望ましい。

5. テストベッドによる実証評価

GICTF[8]が構築するクラウドテストベッドを用いて、IMSに実装した提案方式のセッション情報移動の性能を実測し、実環境においてスケーラブルなセッション情報移動が可能であることを検証した。

テストベッド上に展開されたOpenFlowネットワーク(図5)を利用して、動作検証および性能評価を行った。OpenFlowスイッチによって接続された国内の3拠点に、IMSを構成するCSCFを複数の物理サーバにVMとして構築し、OpenFlowスイッチとL3スイッチの両方に接続した。HSSは、仙台拠点に構築し、L3スイッチに接続した。セッション情報移動制御機能およびバックアップサーバについては、博多拠点に配備した。

CSCFとHSSは、オープンソースのSIPサーバであるOpenIMS[9]を用いて実装した。また、CSCFに具備するOFIMはソフトウェアで実装した。さらに、オープンソースのUEエミュレータであるIMS Bench SIP[10]を用いて多数のUEを模擬した。

本環境において、セッション情報のバックアップがCSCFの処理性能に与える影響(5.1節)、セッション情報の復元の性能(5.2節)、省電力化の効果(5.3節)について検証した。

5.1 セッション情報バックアップの影響

セッション情報のバックアップでは、CSCFによる呼処理の要所（発呼処理を例とした場合60回中23回）で

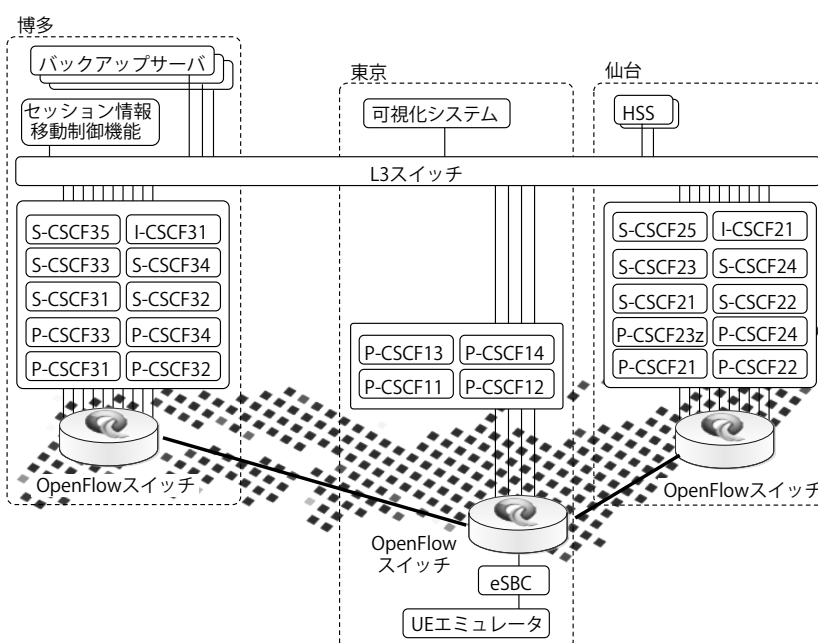


図5 実証に用いたネットワーク構成

セッション情報をバックアップし、このときCSCFは呼処理を待機させる。この待機による呼処理性能の低下の度合いを図6に示す。図は、1台の物理サーバ上に動作させる1つのP-CSCF VMに対して、レートを変えて発呼を行い (x軸) 処理された呼数 (y軸) を示している。呼処理に必要な他のCSCFには複数台の物理サーバを用い、このP-CSCF VMのみがボトルネックになるようにした。

筆者らの実装では、発呼レートが100cps (call per second) までは処理性能が低下しないことが分かる。実用を考えた場合、たとえば1億UEを収容するIMSは、1UEが年平均で1.3呼/日の発呼を行う[11]ことから、平均1,500cpsの性能が求められる。たとえ混雑時に発呼が集中して、発呼レートがこの20倍に達するとしても、300VMでP-CSCFを構成すれば、処理性能の低下を防ぐことができる。

5.2 セッション情報の復元性能

セッション管理を複数のCPUコアに負荷分散することで性能が向上することを実機で確認した。セッション管理に割り当てるCPUコア数を変化させたときのeSBCのスループットとメッセージ処理性能を図7、図8に示す。処理性能は、セッション管理に割り当てるCPUコア数に比例して向上し、5コア割り当て時に発呼処理 (INVITE) で約750Mbps(78,000pps)、UEの登録処理 (REGISTER) で約480Mbps(120,000pps)の性能を達成した。本論文の実装では、CPUコア数の制限から、セッション管理に5コアまでしか割り当てることができなかった。しかし、セッションテーブルは約200,000アクセス/秒の性能を有するため、約10コアまで負荷分散が可能である。より多数のCPUコアを有するプロセッサへ実装することで、5コア分散時に対して約2倍の性能を達成できると考えられる。

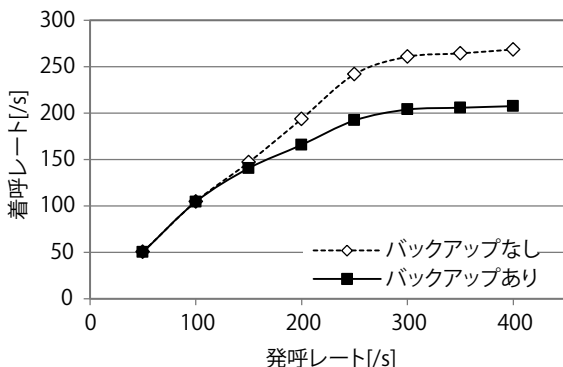


図6 P-CSCFにおける呼処理性能

5.3 省電力化の効果

セッション数の減少に伴って、4台の物理サーバから3台へ削減するときの消費電力の変化を図9に、CSCFの構成変更の周期に対する省電力効果を図10に示す。

図9に示す通り、物理サーバを削減しない場合に約8%しか消費電力を削減できないIMSの実装に対して、構成変更を実施することで最終的に20%の消費電力削減ができる。セッション数減少後の約20秒間は、セッション情報の移動先を決定する最適化計算 (11～13秒)・セッション移動 (14～21秒)・物理サーバの停止 (22～28秒) が順次実行され、セッション情報移動時には消費電力が一時的に増加する。セッション数増加時の物理サーバ増加についても、詳細を割愛するが、サーバ起動時に消費電力の増加が伴う。

図10では、平均通話時間を120秒、平均発呼レートを周期的に変化 (400cps⇔320cps) させた場合の消費電力削減の効果を示している。平均通話時間の2倍以上の周期で構成変更を行うと、5%以上の電力削減が可能になることが分かる。電話サービスの平均呼量は1日の中で大きく変動する (3～4倍) ため、1日レベルの長時間の呼量の変化に対しては図9に見られる大きな電力削減が期待される。また、短時間 (数分) の呼量の変化に対しても、図10に示すように、セッション情報移動の周期を適切に選ぶことで、電力削減が可能である。

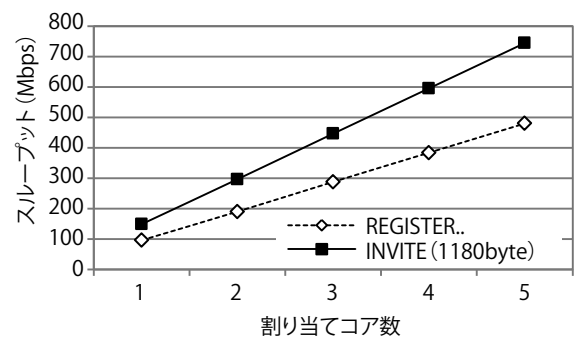


図7 eSBCのスループット

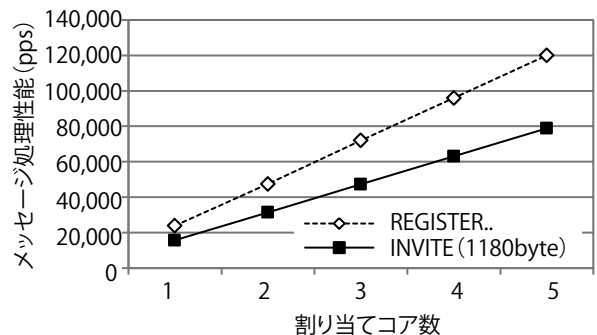


図8 eSBC メッセージ処理性能

5.4 実装に関するプラクティス

実証実験環境の構築では、システム全体の性能向上に向けたボトルネック個所の特定とその対応に時間を要した。当初は、実装工数を最小限にとどめるためにオープンソースソフトウェアを活用し、セッション情報を分散保持するデータベースにMySQL[12]を、ノード情報の収集にZABBIX[13]を用いたところ、両者が性能面においてボトルネックとなっていることが判明した。データベースでは、MySQLが提供するデータの分散保存機能に拠点間の通信遅延が大きく影響することが判明したが、MySQLを直接修正することは実装工数の肥大化を招くために、同機能をCSCFに実装し複数のバックアップ処理を並行に動作させることで遅延の影響を最小限にとどめた。ノード情報の収集機能についても、必要な機能のみを新規に開発することで対応した。

これらのことから、以下のプラクティスが得られた。オープンソースソフトウェアを利用する際には、そのソフトウェアの性能限界および性能向上が容易に可能かなどを把握することが望ましく、実装内容についての知識を深めることが重要である。これにより、システムのボトルネック個所がオープンソースソフトウェアに起因するものか早期に特定でき、さらにそのソフトウェアの改修で対応可能か、もしくはシステムの設計変更が必要か

などの見通しを立てることが可能となる。

また、多数のUEを模擬する際にも実装工数を最小限にとどめるために、オープンソースソフトウェアのUEエミュレータ（IMS Bench SIPp）を活用した。本ソフトウェアの動作環境や動作方法に関するプラクティスを以下に述べる。多数のUEを模擬する際には、IPアドレスとポート番号の組合せがUEの台数分必要となるが、UEエミュレータを動作させるVM数を削減するために、IPエイリアスにより1台のVMに複数のIPアドレスを割り当てた。さらに、UEエミュレータにあらかじめ用意されている基本的な分布（e.g., 指数分布）に従ったシナリオを利用して、現実を想定した発呼レートの変動を模擬するために、それぞれ設定が異なる複数のUEエミュレータを時間差で動作させた。

5.5 性能向上に向けた対策

GICTFテストベッドにおける性能評価では、1) 複数拠点へのバックアップ性能の低下と、2) 通信遅延の影響によるセッション情報のリストアの性能低下が見られた。

1) では、3拠点中2拠点が停止しても残りの1拠点でセッション情報をリストアすることを想定したために、CSCFによるセッション情報のバックアップ頻度が高く

なり、信号処理性能が低下した。このような問題に対しては、冗長度が低下するものの、各拠点から他の1拠点のみへセッション情報をバックアップして、その頻度を下げた。一方で複数拠点の同時障害の可能性を低く抑える対応として、拠点間の通信路が同じ物理回線を通らない構成によって冗長度を高めることが望ましい。

2) では、多数のセッション情報をリストアしかつeSBCのラベル書き換えを行う際に、通信遅延の影響を最小限にとどめる構成が望ましく、同一拠点内でセッション情報を復元するバックアップとその容量設計が求められる。また、拠点規模の多数のセッション情報のリストアは、余剰物理サーバを多数稼働させて容量を確保しておく必要があり、電力削減の効果を低下させる要因になる。そのため、リストアするセッション情報の容量を最小化する設計が重要である。たとえば、 n 拠点がそれぞれ S_k セッション($k=1, 2, \dots, n$)を収容する場合、 $S_m=\min(S_k)$ を満たす

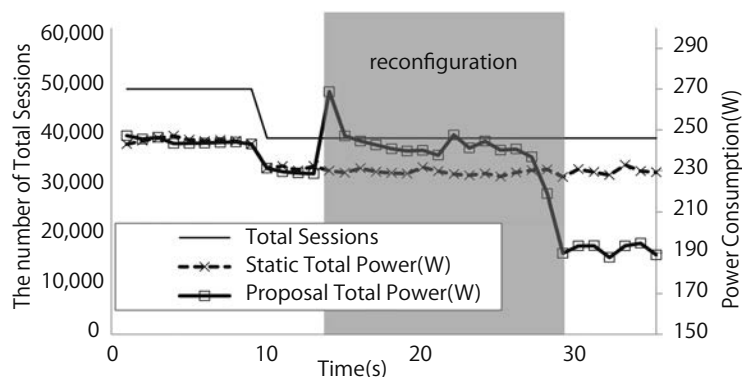


図9 セッション数の減少に伴うCSCF削減時の電力の変化

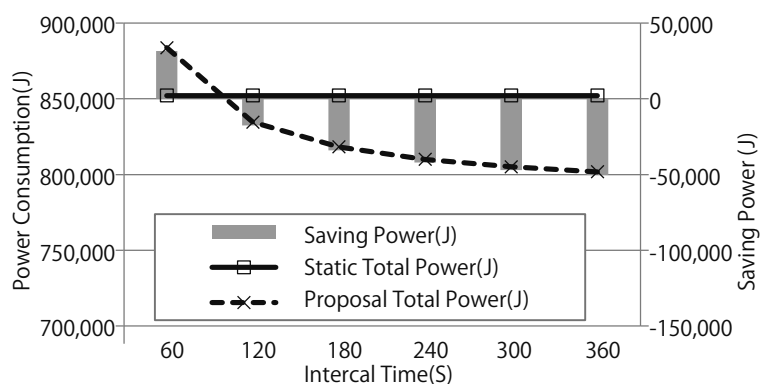


図10 CSCFの構成変更インターバルと省電力効果の関係

拠点 m が他の $n-1$ 拠点のセッション情報を復元し、ある拠点 $j (j \neq m)$ が拠点 m のセッションを復元する。このようにすれば、任意の一拠点のセッション情報を復元でき、復元の容量を含めたIMS全体のセッション情報の容量を最小の $\max(S_k) + \min(S_k) + \sum S_k$ にとどめることができる。

5.6 実験に関するプラクティス

実運用に向けた実験環境構築に関するプラクティスを以下に述べる。実運用を想定した実験データを取得するために、現実を想定した多数のUEからの発呼および発呼レートの変動を模擬する必要がある。実験に必要な発呼性能を確保するために、複数のVM上でUEエミュレータを動作させる必要があったが、実験が容易に再現できるように複数のUEエミュレータを集中制御により操作できるようにした。

実運用時のトラブルシュートの際に正常時の動作および性能確認のためにSIPメッセージのパケットキャプチャデータを参照することがある。IMSは複数のノードによる構成されることから、取得されたパケットキャプチャデータは複数のデータに分割されて存在することとなる。これらのパケットキャプチャデータの整合を取れるように、各ノードの時刻同期とパケットキャプチャソフトの起動・終了を集中制御できるようスクリプトを作成した。

また、システムの動作状況を即座に把握できるように、取得したデータをリアルタイムにグラフ表示することが望ましい。今回のシステムでは物理サーバおよび仮想サーバのリソース状況を一画面に表示するためにksar[14]を用いた。IMSを構成する各ノードが処理しているセッション数および各ノードの消費電力については、専用のGUI画面を開発することでリアルタイムにグラフ表示できるようにした。

6. おわりに

省電力性と信頼性を兼ね備えた移動体通信ネットワークのサービス制御システム (IMS) に関して、サービス無停止でのセッション情報移動方式、利用状況にあわせたIMSの最適な構成を導出する方式、セッション情報のバックアップ・復元の性能を向上する実装方法を示した。また、テストベッドによる実証を通して、同方式の実用性について論じた。

筆者らのアプローチはセッション情報の単位で情報のサーバ収容替えを特徴としており、VMマイグレーション

に比べ、運用面で次の利点がある。一つひとつのセッション情報のメモリサイズがVMのそれに比べて十分小さいため、セッション情報の移動の際にサービスが停止しない。その結果、CSCFの負荷がピークに達したことを確認した後に、リアクティブに収容変更が可能で、電力削減効果が高いということを実証した。また、CSCFに収容される全セッション情報を他のCSCFへ移動できることから、オンデマンドにサーバソフトのアップグレードにとりかかることができ、運用面の効率化につなげることが可能である。

参考文献

- 1) 3GPP TS 24.229 V9.13.0: IP Multimedia Call Control Protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP), Stage3 (2012).
- 2) The Stanford OpenFlow Team: OpenFlow Switch Specification Version1.1.0 Implemented, <http://www.openflow.org/documents/openflow-spec-v1.1.0.pdf> (2011 年 6 月 22 日現在)
- 3) Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and Schooler, E.: SIP: Session Initiation Protocol, Internet Engineering Task Force, RFC 3261 (2002).
- 4) Fakhfakh, M., Cherkaoui, O., Bedhief, I.L. and Frikha, M.: Highavailability in IMS Virtualized Network, First International Conference on Communications and Networking, pp.1-6 (2009).
- 5) Asma, B.L., Amel, H., Maha, J. and Sami, T.: State of the Art and Research Challenges of New Services Architecture Technologies: Virtualization, SOA and Cloud Computing, International Journal of Grid and Distributed Computing, Vol.3, No.4, pp.69-88 (2009).
- 6) Petrucci, V., Loques, O. and Mossé, D.: Dynamic Optimization of Power and Performance for Virtualized Server Clusters, Proceedings of the 2010 ACM Symposium on Applied Computing, pp.263-264 (2011).
- 7) Usui, T., Nishinaga, N., Kitatsuji, Y. and Yokota, H.: Restoring CSCF by Leveraging Feature of Retransmission Mechanism in Session Initiation Protocol, The Third International Conference on Emerging Network Intelligence, pp.50-56 (2011).
- 8) Global Inter-Cloud Technology Forum, <http://www.gictf.jp/>
- 9) The Open Source IMS Core Project, <http://www.openimscore.org/>
- 10) IMS Bench SIPP, http://sipp.sourceforge.net/ims_bench/
- 11) 総務省：トラヒックから見た我が国の通信利用状況【平成 23 年度】，http://www.soumu.go.jp/main_content/000194119.pdf
- 12) MySQL, <http://www-jp.mysql.com/>
- 13) ZABBIX — The Enterprise-class Monitoring Solution for Everyone, <http://www.zabbix.com/jp/>
- 14) ksar, a sar grapher, <http://sourceforge.net/projects/ksar>

伊藤 学 (非会員) mn-itou@kddilabs.jp

2008 年早稲田大学大学院国際情報通信研究科国際情報通信学専攻修士課程修了。同年 KDDI (株) 入社。2008 ～ 2013 年 (株) KDDI 研究所モバイルネットワークグループ研究員。IMS, EPC の研究開発に従事。現在, (独) 情報通信研究機構専門研究員, 電子情報通信学会会員。

小森田 賢史 (正会員) sa-komorita@kddilabs.jp

2006 年東京大学大学院情報理工学系研究科電子情報工学専攻修士課程修了。同年 KDDI (株) 入社。2006 ～ 2013 年 (株) KDDI 研究所モバイルネットワークグループ研究員。SIP, IMS/MMD, WebOS の研究開発に従事。2012 年電子情報通信学会学術奨励賞受賞。現在, KDDI (株) 商品企画部, 電子情報通信学会会員。

北辻 佳憲 (非会員) kitaji@kddilabs.jp

1995 年大阪大学基礎工学部制御工学科卒。1997 年同大学院修士課程修了。同年国際電信電話 (株) (現 KDDI (株)) 入社。現在, (株) KDDI 研究所モバイルネットワークグループリーダー, 博士 (情報学)。コンピュータネットワーク, インターネット, モバイルネットワークの研究に従事。2013 年電子情報通信学会活動功労賞受賞, 電子情報通信学会シニア会員。

横田 英俊 (正会員) yokota@kddilabs.jp

1990 年早稲田大学理工学部電子通信学科卒。1992 年同大学院修士課程修了。同年国際電信電話 (株) (現 KDDI (株)) 入社。1995 年～1996 年米国スタンフォード研究所 客員研究員。現在, (株) KDDI 研究所ネットワーク設計グループグループリーダー, 博士 (国際情報通信学)。コンピュータネットワーク, インターネット QoS, モバイルネットワークの研究・標準化に従事。1998 年電子情報通信学会学術奨励賞受賞, 2005 年情報処理学会山下記念研究賞受賞, 2006 年情報処理学会論文賞受賞, 電子情報通信学会シニア会員, IEEE 会員。

林 偉夫 (非会員) hayashi@ap.jp.nec.com

1994 年大阪大学大学院理学研究科数学専攻修士課程修了。同年 NEC 入社。現在, NEC クラウドシステム研究所主任研究員。データセンタネットワーク, ネットワークセキュリティの研究開発に従事。

省略語

CSCF	Call Session Control Function
eSBC	enhanced SBC
HSS	Home Subscriber Server
IBCF	Interconnection Border Control Function
I-CSCF	Interrogating Call Session Control Function
IMS	IP Multimedia Subsystem
OFIM	OpenFlow Interface Module
P-CSCF	Proxy-Call Session Control Function
S-CSCF	Serving-Call Session Control Function
SBC	Session Border Control
SIP	Session Initiation Protocol
SIP URI	SIP Uniform Resource Identifier
UE	User Equipment

投稿受付: 2013 年 6 月 5 日

採録決定: 2013 年 9 月 9 日

編集担当: 平田圭二 (公立はこだて未来大学)