

通信サービス論理ソフトウェアの処理単位への分割方式の評価

1C-7

野村 眞吾 池 克俊 山田 秀昭 中尾 康二

国際電信電話株式会社 研究所

1. はじめに

種々の多様なサービスの導入の迅速化を目的としたインテリジェントネットワーク (IN) アーキテクチャが検討されている^[1]。IN では、サービス論理 (SL) を汎用ワークステーション上でプロセスとして動作するプログラム (SLP) として実現することが可能である。SL の実現では、呼処理接続中のサービス処理を一つの SLP とし、通話中は連続するサービス処理の単位で SLP とする SLP 分割方式が有効であるとの見通しを得ている^[2]。本稿では、SLP が動作するサービス制御機能部 (SCP) で、SLP 分割方式を提供する機能を実現し、サービス制御要求の受信から呼処理制御命令送出までの時間を測定した結果から、SLP 分割方式の有効性を評価する。

2. サービス論理の分割方式

2.1 SLに基づくサービス処理

サービス交換機能部 (SSP) は、基本呼処理手順 (BCP) に従って、呼のルーティング等の呼処理を実行する。サービスに依存した処理 (サービス処理) が必要になると、サービス制御要求 (IDP) を SCP に送信して、呼処理制御命令を待つ。呼処理制御命令には、呼処理の継続命令 (Continue)、指定したルーティングアドレスへの接続命令 (Connect) などがある。IDP を受信した SCP は、要求されたサービスに対応する SL に基づいて、サービス処理を実行し、SSP に呼処理制御命令を送信してサービスを提供する。サービス処理の途中で BCP の特定の処理ポイントにイベントを設定する命令 (RRBE) を SSP に送信し、BCP 処理がそのポイントまで進み RRBE に対する応答 (ERB) が返された後、サービス処理を再開したり、呼接続情報の通知を SSP に依頼する命令 (CIRReq) を送信し、BCP 処理の終了時に呼接続情報 (CIRRep) を受信したりすることができる。

2.2 SLの SLP への分割方式

SL を SLP により実現する場合には、一つの SL を一つの SLP として実現する形態 (単一 SLP) と、SL を処理の単位で分割して個々の処理を実現した SLP の集合として実現する形態 (複合 SLP) が考えられる。

交換処理においては、呼をルーティングして呼び出し状態となるまでの呼処理 (呼の接続処理) には即時性が要求されるが、呼び出しから応答、通話状態となってから切断されるまでの時間は接続処理に要求される応答時間に比べて一般に大きい。このような交換処理の特徴を考慮して、効率の良いサービス処理を実現するために、接続処理中を一つの SLP により、通話中は複数の SLP に分割して実現する方式 (図 1) を導出した^[2]。

導出された方式の有効性を実証するためには、単一 SLP と複合 SLP のそれぞれの実現形態について、以下

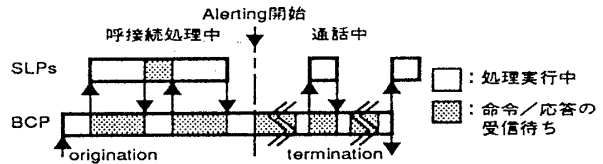


図 1: SLP 分割方式

の比較を行なう必要がある。

(1) SLの分割によるオーバーヘッド

一つの呼に対するサービス処理の実行効率を考慮した場合、SSP がサービス処理の要求を送信してから呼処理制御命令を受信するまでの応答時間を最小とする必要がある。単一 SLP と比較して複合 SLP では、分割した SLP を生成する時間が応答時間のオーバーヘッドとして追加されることとなる。この分割によるオーバーヘッドが、応答時間に与える影響を比較する必要がある。

(2) 並行処理を考慮したサービス処理

単一 SLP では、サービス処理が中断している間においても、プロセスが消滅することなく sleep 状態で存在する。SCP においては、複数のサービス処理が並行して実行されるため、並行に処理する呼数の増加にともなうプロセス数の増加が応答時間に与える影響を比較する必要がある。さらに、同時に受け付け可能なサービス処理要求の数 (多重処理数) を比較する必要がある。

3. サービス応答時間測定実験

3.1 サービス制御機能部の構成

SCP は、SLP の生成を行なう SLP 生成機能部、SSP から受信した応答を対応する SLP に送信する SLP アクセス管理部、生成された SLP の各モジュールからなる (図 2)。それぞれのモジュールは、UNIX 上のプロセスとして動作し、モジュール間の命令、応答の送受信はソケットを用いて実現した。

SLP アクセス管理部は、SSP から IDP を受信すると、SLP 生成機能部に SLP の生成を要求し、以降 SLP からの命令を SSP に送信したり、SSP からの応答を SLP に直接送信する。複合 SLP の制御を行なう場合には、SSP からの ERB や CIRRep などの応答を SLP 生成機能部に送信して、対応する SLP の生成を依頼する。

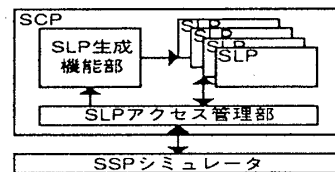


図 2: 実験環境の構成

3.2 サービス応答時間の測定

測定に使用した SL は、IDP を受信してサービス処理を開始し、RRBE, CIRReq, Continue を送信して (命令 1)、次に ERB (応答 1) を受信すると Continue を送信し (命令 2)、最後に CIRRep (応答 2) を受信してサービス処理を終了するものとした。ここでは、SLP の分割

“Evaluation of a Construction Method of Service Logic by the suitable set of Service Logic Programs in IN”
Shingo NOMURA, Katsutoshi IKE, Hideaki YAMADA and Kouji NAKAO
KDD R & D Laboratories

によるオーバーヘッドが応答時間に与える影響を調べるので、サービス処理の内容は単純化した。このSLに対応して、図3に示すようにすべてを一つのSLPで行なう単一SLPと、要求、応答1および応答2の受信により生成され、命令1、命令2の送信およびサービス処理の終了をを実行してプログラムを終了するSLPからなる複合SLPを用意し、応答時間の測定を行なった。

応答時間の測定を行なうために、IDPを送信し、SCPからの制御命令に対する応答を送信するSSPシミュレータ(図2)を作成した。SCPとSSPシミュレータは、それぞれ別のワークステーション上に実装して、シミュレータが要求を送信してからすべての命令を受信するまでの時間を応答時間として測定した。

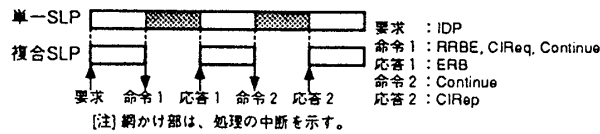


図3: 評価用SLPの構成

(1) プロセス生成時間と通信時間

プロセス生成時間を求めるために、単一SLPに対して、IDPを送信して、命令1を受信するまでの時間(プロセス生成時間+通信時間)と、応答1を送信して命令2を受信するまでの時間(通信時間)を測定した。ここでSLPは、要求、応答1の受信後、即座に命令1、命令2を送信することとした。測定の結果、プロセス生成時間は平均120ms、通信時間は平均10msであった。

(2) 並列処理における応答時間

SSPシミュレータにおいて、同時に複数の要求を送信し、すべての要求に対する命令2の受信を完了するまでの応答時間を、同時に送信する要求の数を変えて測定した。さらに、SLPの処理負荷が応答時間に与える影響を調べるために、応答1を受信して命令2を送信するまでの間に計算処理を追加して、計算処理の時間を5ms, 200ms, 500msと変化させて応答時間を測定した。測定結果を図4に示す。本測定では、応答2を送信しないことにより、単一SLPは命令2を送信した後、消滅することなくsleep状態となるようにした。また、命令1に対しては即座に応答1を送信することとした。

4. 考察

(1) プロセス生成時間と応答時間

要求を受信して、命令2を送信するまでに、単一SLPでは1回、複合SLPでは2回のプロセス生成が行なわれる。図4においても負荷が5msである場合には、同時発生サービス要求数に依存せず常にプロセス生成時間の差が現れている。すなわちSLPを分割した場合、分割数分のプロセス生成時間が処理の遅延となる。従って、一つの呼に対する応答時間を最小とするためには、プロ

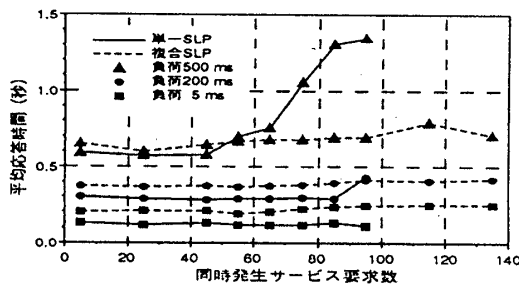


図4: 平均応答時間の測定結果

セス生成の時間を減らすように単一SLPとすることが有効となる。ただし、プロセス生成時間に比べてサービス処理の中断時間が十分長い場合には、複合SLPとしても問題とならない。

(2) 同時に存在するプロセス数と応答時間

SLPの処理負荷が大きい場合は、同時に処理するサービス数が増えると、単一SLPでは応答時間が著しく増加する。計算処理負荷が200msである場合には95個、処理負荷が500msになると50個以上の要求に対しては単一SLPの方が応答時間が長くなる。これは、単一SLPでは、命令2を送信した後、応答2待ちでsleep状態となり、sleep状態のSLPをスワップアウトするための処理が必要となるためである。従って、処理負荷が大きい場合は、一連の処理の単位でプロセスが終了/消滅し、プログラムの分割により処理負荷を分散できる複合SLPとすることが有効となる。

(3) 多重処理の限界

単一SLPでは、システムの制約上95個以上のサービス処理ができない。これは、応答2が受信されるまで単一SLPが終了しないため、プロセス生成が不可能となる95個までのサービス処理が限界となるためである。一方、複合SLPでは、95個以上のサービス処理要求が発生しても、一定の処理でプロセスが消滅するので引き続き95個を超えるサービスの処理が可能となる。

以上のことから、サービス処理の負荷が小さく、同時に処理する呼数が限られる場合には、単一SLPの方がプロセス生成の時間を回避できるという点で有効であり、サービス処理の負荷が大きい場合や、同時に処理する呼数を高める必要がある場合には、複数のSLPに分割する複合SLPが有効となる。

交換処理に対応させると、呼接続処理では、最長でも数百ms程度の即時性を要求されるとともに、中断状態のない連続したサービス処理となるため単一SLPが有効である。また、サービス処理が数秒から数分以上の中断状態となる通話中では、複合SLPを採用してSLPを終了/消滅させることにより、サービスの多重処理が可能となり有効である。従って、筆者らが導出したSLP分割方式(図1)が有効であることがわかる。ただし、呼接続処理中においても、ユーザインタラクションの実行など数秒以上の中断状態が存在する場合や、認証処理などのようにサービス処理の負荷が大きくなる場合は、さらにSLPを分割する必要があると考えられる。

5. まとめ

本稿では、SLを一つのSLPにより実現する場合と、分割した場合の応答時間を測定することにより、SLP分割方式の評価を行なった。その結果から、呼接続処理中のサービス処理を一つのSLPとし、通話中は連続するサービス処理の単位でSLPとするSLP分割方式の有効性を示した。ただし、呼接続処理中においても、サービス処理の中断状態や負荷の大きさを考慮して、さらに分割を行なう必要があることが判明した。本実験では、サービスの処理として単純な計算処理を適用したが、実際のサービス処理を適用した場合について確認する必要がある。最後に日頃御指導頂くKDD研究所 浦野所長、眞家次長、並びに御討論頂いた交換グループ若原リダに感謝します。

参考文献

- [1] ITU-T Q.1200 Series Recommendations, 1993.
- [2] 野村, 菊田, 池, 中尾, "INにおけるサービス論理の処理単位への分割手法", 情処第48回全大, 5D-10, March. 1994.