INにおけるサービス論理の処理単位への分割手法

5D－10　野村 眞吾　菊田 弘之　池 克俊　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、サービス制御機能部 (SCP) がサービス論理 (SL) に従って、サービス交換機能部 (SSP) を制御することによりサービスが提供される。SCPとSSPへの機能分離により、SCPの構築に汎用ワークステーション (WS) を使用して、SCPの機能をソフトウェアとして実現することが可能となった[2]。これによりSLはWS上でプロセスとして動作するプログラム (SLP) として実現されるが、SLをどのような処理単位でSLPとすべきかについてはこれまでに充分な議論がなされていない。本稿では、SLをSLPとして実現する方式の原理を分類整理し、サービス処理の実行効率やプログラムの再利用性の観点から、それらの得失について考察し、この結果に基づいて有効な方式を提案する。

2. サービス処理とサービス論理プログラム

2.1 SLに基づくサービス処理

SSPは、サービスに依存しない基本呼処理手順 (BCP) に従い、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、BCPを停止し、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、BCP処理の再開命令 (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する。サービス処理の結果に応じてSSPに呼処理制御命令を送信して、BCPの処理を再開させる。また、サービス処理の途中で、SCPからBCP中のトリガ検出点 (DP) にイベント (EDP) を設定し、BCP処理がそのDPまで進むのを待った後、SCPのサービス処理を再開したり、BCP処理の終了を待って呼接続情報を処理する場合がある。

2.2 SLPによるSL実現のための要求条件

SCP内で動作するプログラムであるSLPは、SLに基づくサービス処理を実行する。以下に、SLをSLPとして実現するための要求条件を挙げる。

(1) 実行効率

INでは、その呼処理およびサービス処理において、高い実行効率が要求される。特に、汎用ワークステーションを用いてSCPを構築する場合には、SLPを1プロセスに対応付けるため、SCPにおける実行効率に関して以下の2つの側面を考慮する必要がある。

- 一つの呼に対するサービス処理の実行効率
  一つの呼に対するサービス処理の処理時間を考えた場合、サービス処理の要求を受けてから、サービス処理の終了までの処理時間を最小とする必要がある。
- 並行処理を考慮したサービス処理の実行効率
  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。SLPがUNIXなどのOS上でのプロセスとして実現される場合、sleep状態にあるプロセスが多くなると、実行中のプロセスの処理速度に影響がでる。従って、同時に存在するプロセス数を最小とする必要がある。

(2) プログラムの再利用性

INでは、新規サービス導入の迅速化を目的の一つとしており、サービス論理を実現するSLPの開発コストを低くおさえることが要求される。特に、プログラムの再利用性を考慮することが必要である。

3. SLのSLPへの分割の基本原理

3.1 分割の基本原理

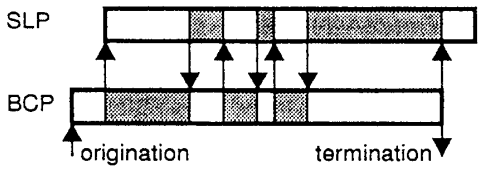
SLをSLPとして実現する場合には、一つのSLPによる実現 (基本原理1) と、複数のSLPによる実現が考えられる。複数のSLPにより実現する場合には、転送機能や番号翻訳機能など機能的な再利用性を考慮した部品 (サービス部品) をSLPの単位とする実現 (基本原理2) と、サービス論理の実行形態を考慮して、サービス処理の開始、および中断後の再開からSSPへ呼処理制御命令を送信して待ち状態となるまでの連続したサービス処理をSLPの単位とする実現 (基本原理3) が考えられる (図1)。

3.2 比較評価

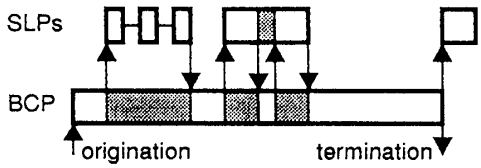
(1) 一つの呼におけるプロセス生成回数に見る実行効率

一つの呼に対する処理のみで見た場合、基本原理1では、プロセス生成が1回であるのに対し、他の方式では複数回となる。プロセスを生成して起動する時間は、sleep状態のプロセスを起動する時間よりも大きくなる

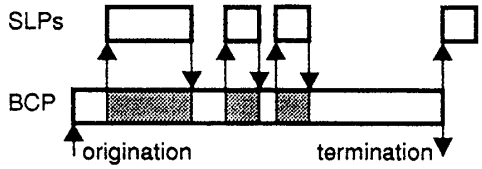
"A Construction Method of Service Logic by the suitable set of Service Logic Programs in IN"
Shingo NOMURA, Hiroyuki KIKUTA, Katsutoshi IKE and Kouji NAKAO
KDD R & D Laboratories



基本原理１：サービス論理単位で1プロセス



基本原理２：サービス部品単位で1プロセス



基本原理３：連続したサービス処理単位で1プロセス

[注] 網かけ部は、処理の中断を示す。

図 1: SLP への分割の基本原理

ので、プロセスの生成回数が多くなると、サービス処理の実行効率は低下する。基本原理２では、サービス部品の粒度を小さくする程、プロセス生成回数が増大し、実行効率が悪くなることになる。

(2) 同時に存在するプロセス数に見る実行効率

同時に存在するプロセス数の観点から見た場合、基本原理１ではサービス処理が開始されてから終了するまでSLP が存在するのに対して、基本原理２および３ではサービス処理が中断している間はSLP が存在しない。従って、複数の呼に対して並行してサービス処理を実行する場合には、基本原理２および３では、同時に存在するプロセス数が最小となる。しかしながら、基本原理１では、sleep 状態のプロセスが存在し、並行に処理する呼の数が多い場合は実行効率が低くなる。

(3) プログラムの再利用

基本原理１では、提供するサービス論理毎に SLP を開発する必要がある。これに対して、基本原理２ではサービス部品毎に SLP とするため、一つの SLP を複数の異なる SL で使用することが可能である。基本原理３では、呼制御命令を送信するまでの一連の処理をまとまりとしているため、SLP の再利用性は低くなるものの、例えば、呼接続情報を SCP 側で管理する処理などは共用が可能であると考えられる。ただし、複数の異なる SL で共用するための汎用的な SLP の設計が必要となる。

4. 分割方式の提案

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。一方、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。例えば、現行の交換機では、呼の接続処理に対して１つのノードで数百 ms 程度の応答時間が要求されるが、呼び出しから応答までは数秒、通話状態では数分以上の安定状態が存在する。

上記のような交換処理の特徴を考慮した場合、接続処理中は基本原理１により、通話中は基本原理３により実現する方式 (図 2) が有効であると考えられる。すなわち、比較評価 (1) により、呼び出し状態 (Alerting) となるまでの呼接続処理に関連する一連の処理を一つの SLP とし、その後は、比較評価 (2) により、連続したサービス処理の単位で SLP に分割する。

具体的には、呼び出し状態の後、No-Answer, Mid-Call, Disconnect などの DP から開始される処理や、BCP 処理が終了して SSP から CIRep (呼接続情報通知) を受信した後の処理をそれぞれ独立した SLP とする。呼接続処理中の SLP は、サービスの完了により終了するほか、上記 DP のみに EDP を設定している状態や CIRep 待ちとなる状態で終了する。

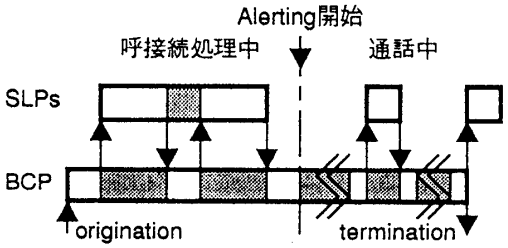


図 2: SLP への分割方式

プログラムの再利用については、SLP の単位で実現することは実行効率上現実的でなく、サービス部品の単位でライブラリ化して再利用性を高めることが有効である。

5. まとめ

本稿では、SL の SLP への分割について３つの基本原理を示し、その得失と交換処理の特徴を考慮することにより、IN に適した SLP への分割方式を議論した。その結果、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 とする方式を導出した。今後、SLP の処理単位に関する定量的評価を進める予定である。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ーダに感謝します。

参考文献

[1] ITU-T Q.1200 Series Recommendations, 1993.

[2] M.Fujinaga, et.al., "An Implementation Method of IN Functional Entities on Top of Distributed Operating System and Its Performance Evaluation Using Experimental System", IEICE Trans. Commun., *E75-B*, No.10, p.1043-1051, 1992.