

## 分散アプリケーションシステムにおける性能管理機能の実装\*

10-1

小澤一友 原田稔

(株) 情報技術コンソーシアム 研究開発部

E-mail: kazutomo@itc.co.jp

## 1.はじめに

システム管理の目的の一つに「利用者に対するサービス性能の維持・向上」が挙げられるが、未だ個々の分散アプリケーションを対象とした性能管理支援は充分でない。本論文では、RPC[1]技術により構築された分散アプリケーションシステムに対する性能管理手法をSNMP[2]管理プラットフォーム上に実装したプロトタイプについて報告する（第3章）。

本手法では、特定の分散APモジュール内の問題、或いはモジュール間の通信の問題へと効率的に問題箇所を切り分けることを考える（第4章）。そのため、各分散モジュールのサービス提供側インタフェース及びサービス要求側インタフェース（以下SI及びCIと呼ぶ、図1を参照）、即ちモジュール間通信の出入口における応答時間などの性能値を、サービス要求の依存関係（第2章）に基づいて管理している。

尚、本論文では、APとは「（システム資源を含まない）アプリケーション」、APシステムとは「APによって利用者がサービスを楽しむシステム（システム資源を含む）」、APモジュールとは「APを構成する（分散した）個々のプログラム」、を意味している。

## 2.分散アプリケーションシステムの性能管理

分散APシステムにおける利用者へのサービスは、複数のAPモジュールや各々の処理が行われる複数ノードのシステム資源の動作に依存している。このため異常時の対応は集中型システムより複雑なものになる。分散APシステムではサービスの依存関係を整理し、システム運用時のサービス性能の問題箇所を効率的に切り分けなければならない。

以下に分散APシステムにおいて想定される依存関係と問題原因の追跡に必要な情報について述べる。

## 2.1 アプリケーションモジュール間の関係

分散するAPモジュール間の関係として以下の2つを想定する。

## (1) 1:1の関係

多層型の分散形態、いわゆるクライアントとサーバの基本的な関係であり、1つのサービス要求モジュールはモジュール間通信を介して1つのサービス提供モジュールに依存する。

異常時には特定のモジュール内/モジュール間通信への問題箇所の切り分けを必要とし、有効な情報として以下を考える。

- ・サービス提供モジュール自身の処理性能 (a1)
- ・モジュール間のサービス要求トラフィック (a2)

## (2) 1:nの関係

並行サーバへの選択アクセスや複数サーバへの順次アクセスに見られる関係であり、1つのサービス要求モジュールがn個のサービス提供モジュールに依存する。1:1の関係が複数の経路に対して結ばれる。

異常時には問題箇所への経路の特定を必要とし、有効な情報として以下を考える。

- ・問題が現れた経路 (b)

## 2.2 アプリケーションとシステム資源との関係

APの処理状態とシステム資源（ノードの計算資源やネットワークの帯域など）の負荷状態などは相互に影響を与え合う関係にある。

異常時の原因分析に有効な、ノード、ネットワーク資源に対するAP側の情報として、以下を考える。

## (1) ノード資源との関係

- ・モジュールの処理状態 (c)

APを構成する各プロセスの処理状態（資源使用量など）は基本ソフト（資源側）が提供する。

## (2) ネットワーク資源との関係

- ・モジュール間通信の入出力量 (d)

\*この研究は、情報処理振興事業協会(IPA)の先進的情報処理技術の開発促進事業で実施されている「広域分散ソフトウェア生産技術開発」プロジェクトで行われた。

A Performance Management Tool for Distributed Application Systems

Kazutomo Ozawa, Minoru Harada

Information Technology Consortium Corp. Research & Development Dept.

Kiba-Kouen Bldg. 5-11-13 Kiba, Koutou-Ku, Tokyo 135, Japan

### 3.分散アプリケーション性能管理機能の実装

第2章に挙げた依存関係を考慮して実装した、性能管理機能のプロトタイプについて述べる。

RPC技術により構築された分散APに対する性能管理機能をSNMP管理プラットフォーム上に実装している。

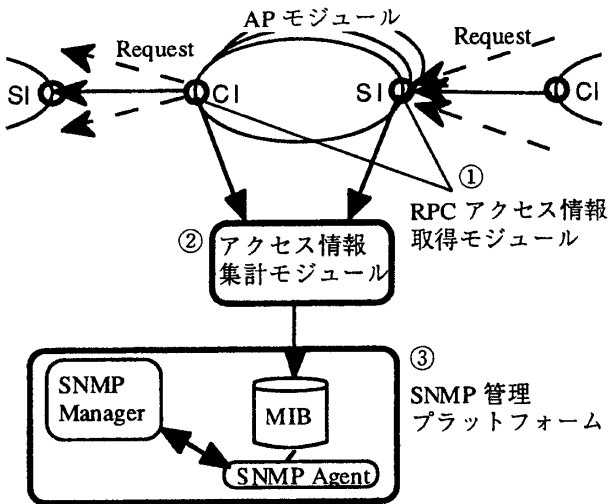


図1 管理対象APとプロトタイプ

#### 3.1 監視ポイント

特に管理するポイントは、各APモジュールのSI・CIの2カ所(図1を参照)であり、ここで主に以下のアクセス情報を取得する。

SI: 処理の開始時刻、サービス要求元、被要求サービス、パラメータサイズ

CI: 要求の開始時刻、要求先、要求サービス、パラメータサイズ

さらに、上記のアクセス情報を集計して主に以下の管理情報を取得する。(())は第2章との対応を示す)

- ・SI及びCIにおける応答時間(a1\*,a2\*\*)
- ・サービス提供・要求の頻度(a2)
- ・問題(応答時間が長いなど)のある関係先(b)
- ・モジュール数(プロセス構成)(c)
- ・入出力データ量(d)

\*a1: CIの応答時間-SIの応答時間

\*\*a2: SIの応答時間-上位CIの応答時間

#### 3.2 プロトタイプの構成

本プロトタイプは、図1に示すように以下の3要素から構成される。

- ①RPCアクセス情報取得モジュール  
チェックポイントにおいて管理対象APからアクセス情報を取得する。
- ②アクセス情報集計モジュール  
アクセス情報取得モジュールから渡されたアクセス情報を集計し、管理情報を生成する。
- ③SNMP管理プラットフォーム  
生成された管理情報をMIB[2]の管理項目と

して収集し、管理作業の支援を行う。

### 4. 問題箇所の切り分け実験

利用者へのサービス性能に異常が現れた際の問題箇所の切り分けをシミュレートした。利用者へのサービス性能とはフロントエンドのAPモジュール(以下、CMとよぶ)のユーザインタフェース即ちSIにおける応答時間である。問題箇所の切り分けに際し、特定箇所の問題はその依存元に伝搬すると考え、問題箇所に向かって追跡を行う。

CMのSIは異常値、CMのCIも異常値。

∴CM外の問題である。

さらに異常値を示す依存先である別ノード上のサービス提供モジュール(以下、SMと呼ぶ)に遡る。

CMのCIは異常値、SMのSIも異常値。

∴CM-SM間の通信の問題でもない。

SMのSIは異常値、SMのCIは正常値。

∴SM内の問題である。

よって当ノードにおいてSMと関係するシステム資源へと原因追及を進めることができる。この時の各SI・CIにおける応答時間の変化を図2に示す。

実際にこの実験ではSMが常駐するノードのCPUに管理対象AP以外の負荷を加えている。

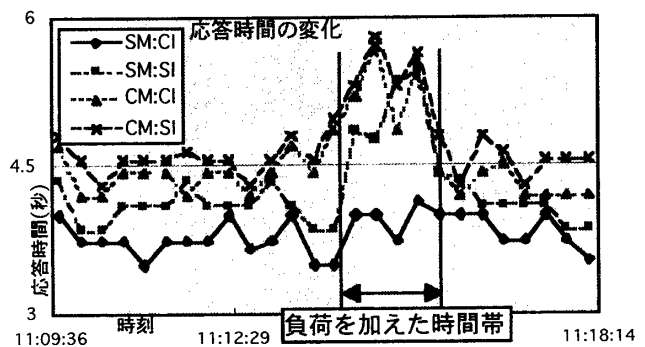


図2 各監視ポイントにおける応答時間の変化

### 5. おわりに

本報告では、分散アプリケーションシステムにおけるサービス要求の依存関係に基づいた性能管理機能の実装について述べ、問題箇所の切り分けに際し有効に動作する一例を示した。

今後は本機能で可能となった問題箇所の追跡・切り分けを自動化し、性能異常の監視及び対応の効率化を計る予定である。

#### 参考文献

- [1]Open Software Foundation, Inc.:Introduction to OSF DCE,1992
- [2]Marshall T. Rose:The Simple Book[An Introduction to Internet Management],1994