

リソース指向分散処理環境の実装及び評価

今井 功, 谷林 陽一, 佐藤 文明, 中川路 哲男, 水野 忠則

3W-4

三菱電機(株) 情報電子研究所

1 はじめに

我々は、処理分散と資源管理を統合した分散処理環境として、リソース指向分散処理環境 (RODS: Resource Oriented Distributed System) を提案し、これまでその計算モデル [1]、実現機構 [2] およびオブジェクトの実現方式 [3] についての検討を行ってきた。RODS は、分散環境として位置透過性、負荷分散、システムの柔軟性や障害透過性などの機能を有している。また、RODS の環境を管理制御するための環境シェルも備えている。プログラマに対しては、オブジェクト指向インタフェースを提供している。

本稿では、RODS の UNIX 上への実装を行ない、その性能の評価を行なった結果について報告する。

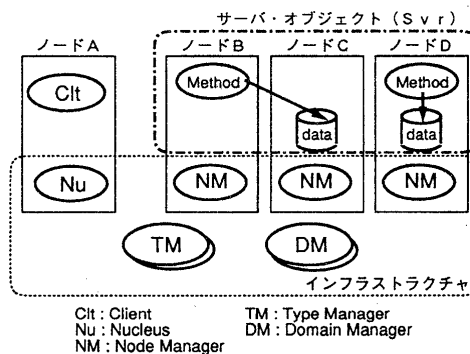


図 1: RODS の構成図

2 動作概要

RODS の構成図を図 1 に示す。RODS はインフラストラクチャとして、ニュークリアス (Nu)、タイプ・マネージャ (TM)、ドメイン・マネージャ (DM)、ノード・マネージャ (NM) の各マネージャが起動されており、この環境上でクライアント (Clt) や、複数のノードにまたがってサーバー・オブジェクト (Svr) が動作している。

以下に、RODS 上でのオブジェクトの生成・消滅とオブジェクトの呼び出しの動作概要について述べる。

- 生成
 

Nu が、クライアントからの要求を受けとり、TM, DM への問い合わせの結果から、生成に最適なノードを決定し、ここで選択されたノードの NM がサーバを起動する。
- 呼び出し
 

Nu がクライアントの要求を受けとり、DM に負荷情報の問い合わせを行ない、最適なサーバを決定し、そのサーバに対して要求を送る。サーバは、自分の動作状態を NM に通知し、NM は負荷情報に変化があった場合、DM に負荷情報を送る。
- 削除
 

Nu がクライアントの要求を受けとり、DM にオブジェクト情報を問い合わせ、サーバの存在するノードの NM に削除要求を送り、サーバを削除する。

Implementation and Evaluation for Resource Oriented Distributed Systems  
 Isao IMAI, Youichi TANIBAYASHI, Fumiaki SATO, Tetsuo NAKAKAWAJI, and Tadanori MIZUNO.  
 Computer & Information Systems Laboratory, Mitsubishi Electric Corporation.

3 性能評価

3.1 測定項目

本評価では、RODS において重要と思われる、以下の項目について性能測定を実施した。

- (1) オブジェクトの生成・消滅時間
- (2) クライアントからのサーバ呼び出し時間
- (3) サーバ・オブジェクトの処理効率

(1),(2) は、クライアントとサーバの数を 1:1 にして測定した時、各マネージャの配置が性能にどのように影響するかを評価する。また (3) では、クライアントとサーバの数を変化させることにより、サーバの応答時間がどのように変化するかを評価する。

3.2 測定条件

本評価では、同一機種の EWS を 2 台使用した。測定項目 (1) と (2) における各コンポーネント (クライアント、サーバ、及び各マネージャ) の配置方法を、表 1 に示す。尚、Clt と Nu は、ノード 1 に常駐しているものとする。また、測定項目 (3) において、評価に用いたクライアントの数とサーバの冗長度の組合せを表 2 に示す。

3.3 測定結果

表 3 に、測定項目 (1),(2) の結果を示す。これは、コンフィグレーション (1) の値を基準とした時に、各コンフィグレーションと相対比較した結果である。また図 2 に、測定項目 (3) の結果を示す。横軸はサーバでの処理時間を、縦軸はクライアントから呼び出した全体の応答時間を各々表しており、表 3 同様に相対比で表している。

表 1: 構成要素の配置

	ノード1				ノード2			
	Svr	NM	TM	DM	Svr	NM	TM	DM
a	○	○	○	○				
b	○	○					○	○
c	○	○	○					○
d	○	○		○			○	
e			○	○	○	○		
f					○	○	○	○
g			○		○	○		○
h				○	○	○	○	

表 2: クライアント/サーバの個数

番号	クライアント	サーバ
X	1	1
Y	2	1
Z	2	2

表 3: 各コンフィグレーション毎の測定結果

測定項目	a	b	c	d	e	f	g	h
生成・消滅	1	0.97	0.97	0.97	0.95	0.92	0.93	0.93
呼び出し	1	0.99	1	1	0.81	0.79	0.80	0.81

- (1) 表 3 におけるオブジェクトの生成・消滅時間をみると、TM と DM の配置位置を換えて測定した結果は、ほぼ同じ値となっている。すなわち、マネージャの位置が、生成/消滅の処理時間にほとんど影響を与えない。
- (2) クライアントからのサーバ呼び出し時間では、クライアントとサーバが別ノードに存在する方が処理速度が速い。
- (3) 図 2 から分かるように、ある程度の通信のオーバーヘッドが伴うが、サーバでの処理時間の増加に応じて、クライアントでの応答時間が、直線的に増加している。クライアントが 2 つになると、その応答時間はほぼ 2 倍になり、サーバも 2 つにすると、応答速度は 1:1 の関係の値に近付いている。

#### 4 考察

測定結果 (1) の原因は、データを伴わない測定のため、通信機構自体が、それほどローカルとリモートの通信に差がなかったためと考えられる。また、プロセスの生成時間が大きいので、通信時間の影響が少なかったと考えられる。

結果 (2) の原因としては、クライアントとサーバの位置関係の違いにより、2 つの最もアクティブな Nu と NM の位置が分散され、間接的に速度が向上したのと考えられる。

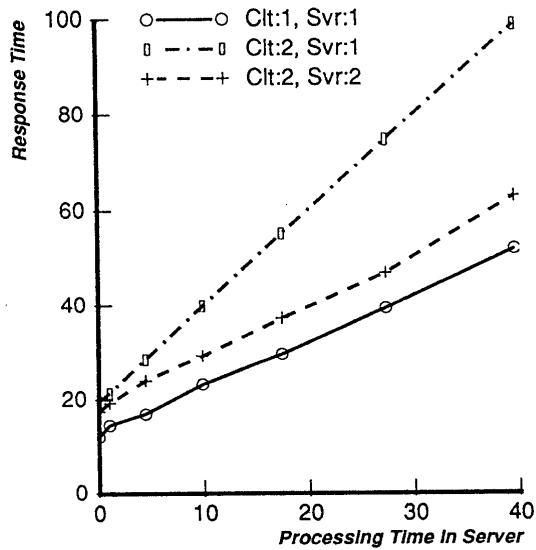


図 2: クライアントとサーバの処理時間

測定結果 (3) のクライアントとサーバの数が、2:1 から 2:2 になった際に、応答時間が約半分になったのは、RODS の動的な負荷分散機構が有効に働いたためと考えられる。但し、クライアントとサーバが 2:2 の場合、1:1 の場合よりも多少の応答時間の増加が見られる。これは、RODS の持つ動的な負荷分散機構において、負荷情報通知の遅れによって、負荷の分散が適切に行なわれない場合があり、応答時間の増加に繋がっていると考えられる。

#### 5 おわりに

本稿では、リソース指向分散処理環境の性能評価について報告した。今後の課題としては、更に計算機の台数を増やした場合や、個々のマネージャ内の処理時間についての評価を行なっていき、問題点の検出やその改善を行なっていく予定である。

#### 参考文献

- [1] 谷林 他: 処理分散と資源管理を統合した分散処理環境の提案 (1) - 計算モデル -, 第 43 回情報処理学会全国大会 (1991).
- [2] 中川路 他: 処理分散と資源管理を統合した分散処理環境の提案 (2) - 実現機構 -, 第 43 回情報処理学会全国大会 (1991).
- [3] 谷林 他: 分散処理環境におけるオブジェクト実現方式, 第 44 回情報処理学会全国大会 (1992).