ネットワークを介した
分散型エンジニアリングデータベースシステムの開発

大野　尚則*　　棚橋　英樹*

A Development of a Distributed Database System for Network Engineering

Naonori Ohno* and Hideki Tanahashi*

あらまし　近年，製品開発は大量生産から多品種少量生産や多品種変量生産へと急速に移行し，製品開発の高速化や効率化のためにディジタルデータの再利用の必要性が高まっている．本報告では，設計から生産までの工程を，分散型データベースシステムを用いることによりネットワークを介して連携する方法を提案する．提案する方法は，生産活動の各作業工程に利用されている現存の CAD システムや加工機などと小規模なオブジェクト指向データベースシステムを連結させ，これらのデータベースをその上位となるデータベースがネットワークを介して管理する方法である．本手法を適用したデータベースシステムを試作し，現在一般的である一極集中型データベースによるデータ管理方法とクライアント数による性能変化への影響について比較した結果，提案した方法はシステムの性能劣化が少ないことが検証された．

キーワード　分散データベース，生産システム，エンジニアリングデータ，分散オブジェクト

１．　まえがき

近年インターネットの普及により，WWW や電子メールなどのアプリケーションが，地域を越えたコミュニケーションのための必要不可欠なツールとなった．工業分野においても大企業ではイントラネット等を用いた大規模データベースによるデータの集中管理が行われるようになり，更には CALS に代表される情報基盤の整備が進んでいる．しかしながら，中小企業においてネットワーク技術は，コストの問題や所有する機器のネットワーク対応の遅れ等により，未だ積極的に活用されておらず，生産活動自体に有効に生かされていないのが現状である．

一方では CAD/CAM/CAE システムの普及により様々なデータが電子化されているが，互いのデータが有効に利用されていない．これは，前述したネットワークに関する問題に加えて，システムの違いやシステム間で扱うデータの互換性の問題が原因と考えられる．

これらをまとめると，中小企業におけるネットワーク技術利用に関する問題とデータの互換性に関する問題に大別できる．現在，製品の多品種少量生産だけにとどまらず，個々の仕様に応じた製品開発の必要性が高まっている．したがって更なる製品開発の効率化のためには，設計から生産までをネットワーク上で行い，各工程で生成されたデータを積極的に利用していくことが重要となってくる．

また，ネットワーク技術利用に関する問題を解決するためには，現場の作業者にネットワークを意識させないシステム開発や優れたユーザインタフェースが必須であり，これらの実現により，迅速なデータ蓄積，管理，共有が実現でき，作業の効率化が期待できる．更にデータの互換性に関する問題については従来までの IGES や DXF などのデータ形式から，STEP や XML などのより広い領域を定義できるデータ形式が主流になりつつあり，これらのデータ形式にも対応しておくことも必要である．

本研究は，中小企業庁 地域活性化補助事業の共同研究事業の一環として，中小企業の情報基盤の整備を目的とし工業技術院 機械技術研究所と国内４カ所の公設研究所と共同で，ディジタルエンジニアリングデータの利用や再利用技術の確立を目指している．

本報告では，その部分要素となるネットワークを介して生産活動に関するエンジニアリングデータを共有するための分散型データベースシステムについて，設計・生産現場における本システムの有効に活用するための方法について述べる．

２．　ネットワーク型の設計・生産方法

２．１　製品開発の現状

中小企業が独自の製品を開発する場合における生産活動について考えると，中心となる企業が異なる分野の技術を有する複数の企業と連携することが多い．この場合，図面やデータの授受は，担当者を介した手渡しや FTP 等

*情報システム部
Information Division

を用いたネットワーク経由で行われており，設計変更や修正ごとにデータの授受が行われる．この方法の問題点として，

・データの管理はデータを渡された作業者にゆだねられているため，企業として管理ができていないことが多い，
・渡されるデータは必要最小限である，
・全体を通してデータの流れの把握が容易ではない，
・複数のプロジェクトを抱えた場合，管理が難しい，

などが挙げられる．

中小企業が連携をとって製品開発を効率的に進める場合には，それぞれが単独に作業を行うのではなく，各企業が大企業の１部門のように，更に密に連携し，開発対象に関連する互いのシステムやデータ等のリソースを共有して作業を進めていくことが望ましい．特に，作業中に発生した様々なデータを電子化されたデータとして残すことは，生産の効率化という点では必要不可欠であり，これらのデータは新たに製品開発を行う場合に生じる問題の解決につながる．図１はネットワーク対応型の設計・生産システムの一般的な例である．

２．２　ネットワーク連携の問題点とその解決策

図１に示すシステムの運用を考えた場合，

１）企業や工程ごとに使用するシステムが製品やバージョンの違いにより多岐にわたり，すべての情報を共有することが容易でない，
２）データを集約して共有化する場合，大規模なデータベースサーバが必要となる上，企業のノウハウや機密情報を含むエンジニアリングデータの管理がサーバの管理者に委ねられセキュリティに問題が残る，
３）データには，設計データをはじめ，画像や音声データなどマルチメディア的な要素が含まれ，現在一般的に利用されているリレーショナル型データベース(以下 RDB: Relational DataBase) には適していない，

などの主に３つの課題が残されている．

課題１についての解決策としてプロジェクトに携わる企業が利用するすべてのシステムとそのバージョンを統一化する方法が最善であるが，コストの面で実質的に困難である．多種多様なデータを効率的に管理する場合，実用的な面から考えられる解決策として PDM（Product Data Management）システムを用いたデータ管理方法があげられる．PDM システムは主に設計データの版管理などに利用されており，大規模なプロジェクトでは必要不可欠なシステムとなっている．

課題２はセキュリティに関するもので，データが社外のサーバにある場合，必ず生じる問題である．企業にとってはノウハウを含んだ重要なデータに関しては社内で管理し，必要に応じて提供し共有する方法が望ましいと考えられる．これを実現するためには各企業に小規模なデータベースを設置し，これらを連携することによって，データを共有する方法が必要である．

課題３については，マルチメディア的な要素を多く含んでいるエンジニアリングデータには一般的に、２次元配列の集合である RDB よりもオブジェクトの集合であるオブジェクト指向型データベース（以下 ODB：Object-oriented DataBase）が適している．RDB はデータ構造の特徴からエンジニアリングデータや XML のように構造化されたデータ形式の格納には適していない．

これらの問題に対して，我々はデータベースとして ODB を採用し，エンジニアリングデータや構造化されたデータの格納を可能とする．また，各工程ごとに専用の小規模なデータベースを設置することにより各工程でのデータ管理が可能となりセキュリティに関する問題も減少する．更に，これら複数のデータベースを、ネットワークを介して連携させ，かつ各工程で利用される機器やシステムに密着したデータベースを構築することにより，データベースを介した統合的な生産システムが構築できる．

これまでにも生産システムのオブジェクト化によるネットワーク分散に関する研究[1]〜[3] が行われているが，NC (Numerical Control)加工機[1][2]や PLC (Programmable Logic Controller)[3]など生産工程周辺に限られている．

本論文では，設計から生産までを対象としたネットワーク型の生産システムを目指し，その根幹となるエンジニアリングデータのための小規模分散型データベースシステムを提案する．

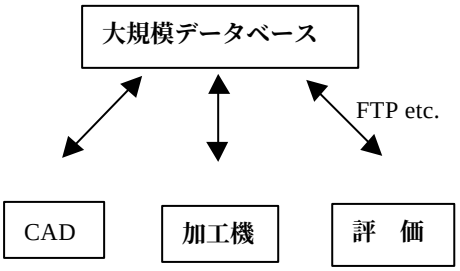


図１　一般的なネットワーク連携

Fig.1 A group work through network

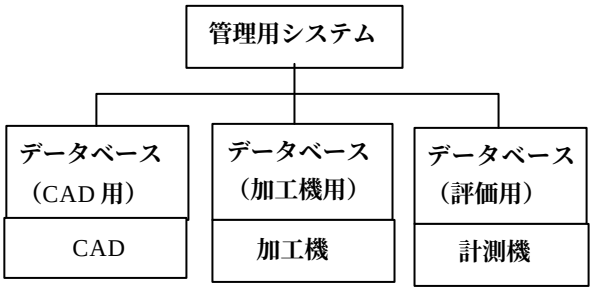


図２　提案するネットワーク連携

Fig.2 An ideal group work through network

３．　小規模分散型データベース

本章では，まず 3.1 節でエンジニアリングデータを複数のデータベースに分散して管理する方法について概説する．次に，この方法を実際の設計・生産工程に適用する場合の可能性について，想定される適用例と併せて述べ，更にシステム内でのデータの流れを明確にする．

3.3 節では, この提案方法を実現するために必要となる要素技術について概説した上で，簡易型の小規模分散型データベースシステムの実装について述べる．

３．１　分散管理手法

2.2 節にあげた問題を要約すると，設計・生産活動における多種システムの統合化，データベースの管理方法，データベーススキーマの３点に集約される．これらの問題を解決できるであろう理想的なシステムのブロックダイアグラムを図２に示す．まず設計工程を考えた場合，概念設計，詳細設計，解析の各工程が互いにセキュリティが保たれた状態で，ある条件を満たす場合にのみ内部データが開示され，利用できる．それぞれの工程では作業者の作業履歴や実験的に作成された情報など極力，データベースに格納する．この格納作業を作業者が介さず自動的に行うことが理想的である．さらに今後の再利用のためには，必要に応じて迅速にその情報を取り出せなければならない．

また生産工程や検査工程では，加工機や検査装置から出力される結果や信号がデータベースに自動的に格納され，設計工程への再利用や，装置自体の状態の把握に用いられる．このように一連の作業が生産活動全体を通して実現できれば，異なる企業同士の連携をより効率的に行うことが可能となる．

しかしながら，図２に示すデータの自動格納・読み出し型の生産システムを実現するためには，各工程のシステムとデータベースが直結していることが前提であり，CAD などの市販のシステムを利用している場合，データベースとシステムの直結は難しい．市販システムはシステム内部に関わる部分を改良することが不可能であるからであり，使用システムに応じてデータベースとシステムとの連携について検討する必要がある．

このことから本研究では生産活動におけるエンジニアリングデータの連携モデルを 3 つのパターンに分けて検討する．

３．２　設計・生産工程への適用についての検討

本節では，市販システムのような閉システムとの連携を疎な連携モデル，自社開発や情報が開示されているシステムとの連携を密な連携モデル，プロジェクトマネージャなど第３者がデータを閲覧する閲覧モデルの 3 つに分けて，それぞれに対して取り得るアーキテクチャを示す．生産活動に利用するすべての機器，システムは３種類のどれかのモデルに属する．また，これらのモデル内で扱うデータベースはマルチクライアント型とする．

１）疎な連携モデル（図３（a））

市販システムから出力されたデータは，手作業または半自動的にデータベース GUI を通して，Local に設置されたデータベースに格納された後，ネットワークを介して他の工程に利用される．この場合，データベース GUI とデータベース本体は一体化して開発することもできるが, オブジェクト指向技術の MVC モデル[4]に基づいて，分離して開発する方法が妥当である．

２）密な連携モデル（図３（b））

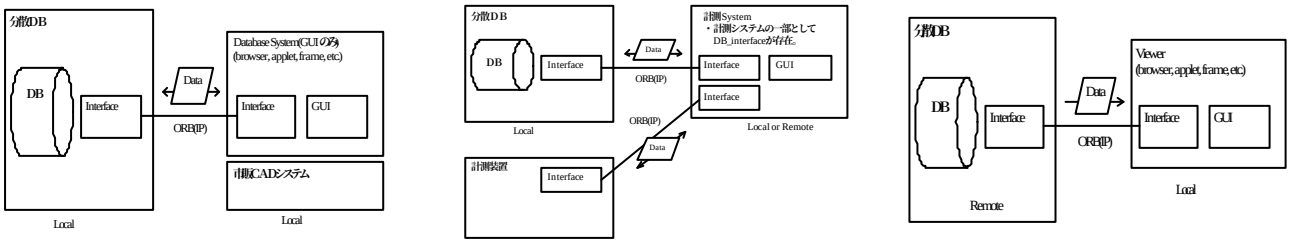
加工機や測定機を対象にした場合には，その制御やデータの I/O の情報が提供されることが多く，独自の制御用ソフトウェアを作成することが可能である. この場合，そのソフトウェアの GUI にデータベース用の入出力部分を組み込むことによって，出力されたデータが自動的にデータベースに格納，読み出しが可能になるだけでなく，データの管理も自動的に行うことができる．

３）閲覧モデル（図３（c））

データベースへのデータ格納を行わない閲覧者を対象としたモデルである．閲覧ソフトウェアを介してデータベースにあるデータを読み出す．したがって，データベース側は複数のクライアントに対応させることが必須である．

３．３　実装

前節までの考察を踏まえて，これらを実現するための基盤として利用するために，図 4 に示す分散型データベースシステムを試作した．機能としてはデータベースシ



(a)疎なモデル
(a) The separate model.

(b)密なモデル
(b) The close model.

(c)閲覧モデル
(c) The browsing model.

図３　連携モデル

Fig.3 The group work models

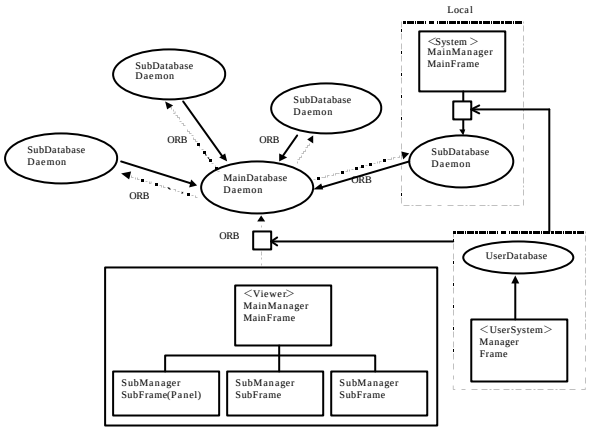


図４　試作分散型データベース

Fig.4 The prototype system

ステムに最低限必要なデータの格納，読み込み，テキストによる検索機能を持っている．

開発言語には Java 言語[4], 分散オブジェクト管理には ORB（Object Request Broker）用いて HORB[5]を採用した．また小規模な分散データベースを構築するために，アプリケーション組込用のオブジェクトストレージライブラリである PSE pro[6]を用いた．開発に利用した環境を表１に示す．

１）HORB

ORB では OMG（Object Management Group）の規格である CORBA（Common Object Request Broker Architecture）がデファクト標準となっているが，本研究では複数の ORB サーバを利用するため，最も高速な HORB を採用した．

HORB の主な特徴を下記に示す．

・他の ORB よりも２倍以上高速である．

・IIOP（Internet Inter-ORB Protocol）をサポート．

IIOP を採用しているため，CORBA との相互接続も可能である．

２）ODB

前述したようにエンジニアリングデータは，データの形式や構造が様々であるため，RDB には適していない．

また，マルチクライアントに対応した ODB が複数必要となることから，コスト面を考え PSE pro を採用した．本製品はオブジェクトの格納やデータベースとしての基本的な機能をもつ ODB 用のライブラリである．主な特徴は次の通りである．

・アプリケーション組込用である．

・独自のシリアライズ機能[6]を持つ．

４．　システム評価

試作した分散型データベースの性能を評価するために，２種類のデータベースモデルを構築し，データ処理の時間を計測することによって基本性能を比較する実験を行った．実験に用いたデータベースモデルを図５に示す．

表 1　開発環境

Table 1 The development environment

開発環境	VisualCafe 3.0
JavaVM	version 1.2
Database Library	PSE pro Ver.6.0
ORB	HORB 2.0

図５(a)をネットワーク分散型，図５(b)をシステム内分散型とする．ネットワーク分散型は，４台の PC 上にそれぞれ異なるデータベースサーバを構築し，これらのサーバを図中の CoreDB で記述されているデータベースサーバが別の PC 上で管理する（以下 CoreDB とする）．これらのデータベースに対して，クライアントは図中の Viewer と記述されているデータベース GUI を通してアクセスする．

それに対して，システム内分散型は，現在一般的に利用されている一極集中型のデータベースを擬似的に再現したモデルと見なすことができる．１台のＰＣ上に，ネットワーク分散型で使用している４つのデータベースサーバと CoreDB を構築している．クライアントは同様に Viewer を介して，別の PC 上に存在するデータベースにアクセスする．

実験条件として，データベース処理に用いたデータは ID 番号，名称，属性，コメントといったエンジニアリングデータの基礎となるパラメータのみである．またデータベース構造は HashTable[7]を用いる．データ量に関してはすべての実験において，各データベースサーバの扱うデータ量の総計を４千件に設定し，これを格納，読み出しする時間を計測した．システム内分散モデルの場合は，１台の PC で｛４千件×４＋CoreDB の処理｝に要する時間を計測する．

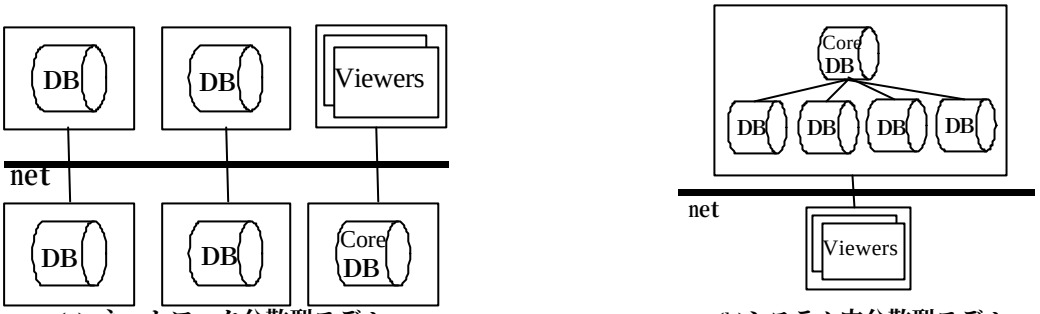
また他のネットワークからのトラフィックによる影響を防ぐため，物理的に接続されていないセグメントで実験を行っている．

４．１　予備実験

データベースモデルの違いによる性能比較実験を行う前に，CPU 性能など PC 性能によるデータベース性能の影響を把握するために，ネットワーク分散型モデルを用いて PC の性能の違いによる性能評価を行った．使用した６台の PC の性能を表２に示す．表中の数値は HDBENCH[8]を用いて計測した値である．

これらの PC を用いて，それぞれに４千件のデータを格納した時間を図６に示す．横軸はトランザクション回数，縦軸は格納時間である．トランザクションとはデータベースが処理する単位であり，この実験では４千件のデータを 20 回（１回につき 200 件）のトランザクション処理で格納している．

結果より，CPU，メモリ，ハードディスク性能に比例して，データベース性能が向上していることがわかる．また，データ数による格納時間推移の傾向は，PC 性能に



(a) ネットワーク分散型モデル
(a) A network distributed model

(b)システム内分散型モデル
(b) A distributed model in a system

図 5　実験に用いたモデル
Fig.5 Two models for the evaluation of the prototype

表 2　実験に使用した PC
Table 2 The list of PC in the experiment

	CPU, 周波数 (MHz)	メモリ(MB)	INT	FLT	MEM_READ	MEM_WRITE	HD_READ	HD_WRITE
system0	P3,600	256	27206	30736	9877	10901	4172	3035
system1	P2,400	128	16053	16885	7865	10232	9714	6294
system2	Ce,333	256	13409	14088	6802	5987	13662	11375
system3	P,75	24	2862	2130	262	1636	1334	340
system4	MMX,120	32	4595	3523	373	2222	3568	943
viewer	P2,400	298	18100	20450	6347	7879	9784	4319

（性能数値は文献[8]参照）

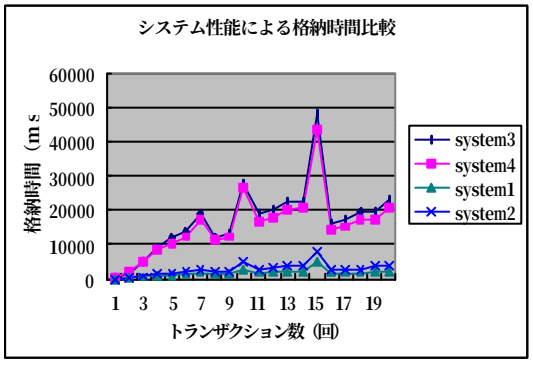


図 6　予備実験結果
Fig.6 The results of the preliminary experiment.

は影響しないことも確認できた．

4．2　データベース性能評価

次にデータベースサーバのクライアント（Viewer）の数（1～4）による各データベースモデルの性能への影響を確認する．ここでは HORB による通信時間を含まないデータベースのトランザクション処理に要する時間を計測した．その結果を図 7 に示す．但し，ネットワーク分散型の実験で利用した PC の内，予備実験の結果から性能が著しく低い PC のデータは省いている．システム内分散型モデルと比較してネットワーク分散型モデルはデータ格納性能についてクライアント数の影響が少ないことが確認できた．しかしながらデータ読み出し性能はシステム内分散型モデルの方が優れていた．

4．3　ORB を含んだ性能評価

最後にクライアントからコマンド（データ格納命令）を発行してから，サーバより結果が得られる間の時間を測定し，4.2 節と同様，クライアント数によるデータベース処理性能への影響を確認した．その結果を図 8 に示す．

4.2 節の実験と同様，データ格納性能はネットワーク分散型モデル，データ読み出し性能はシステム内分散型モデルが優れていた．

4．4　実験のまとめ

予備実験も含め 3 種類の実験を行った結果をまとめると，サーバ性能が PC レベルの場合，クライアント数の増加によるシステム性能の劣化が顕著に現れることが確認できた．特にデータ格納作業についてはデータベースでの処理が直列化され，システム内分散型モデルの場合には処理待ちのプロセスが膨大でシステム資源を浪費するために性能劣化が大きくなっていると考えられる．データ読み出し作業については，データベースシステムが複数のプロセスを同時に処理できるため，システム内分散型モデルの方が ORB による通信がシステム内で行われているために処理時間が少ないと考えられる．

5．　おわりに

近年のネットワークに関連する技術の進歩は飛躍的に進んでいる．ネットワークアプリケーションの開発用言語として Java 言語は認知され，更なる改良が加えらている．また ORB 技術もネットワークプログラミングを容易にする技術として必要な機能を持っており，今後さらに一般的に利用されるようになる．

本研究ではこれらの技術を利用し，更なるシステム統合を計画している．試作した分散型データベースシステムを基盤としてネットワーク統合型生産システムの実現を目指し，簡易的な PDM の機能や各作業工程とのインタフェースの検討を行う．更に具体的な製品を対象にした実証実験を行い，提案手法の有効性を確認する予定である．

謝辞　本研究を遂行するにあたり，工業技術院 機械技術研究所 物理情報部 小島部長をはじめ，本事業に携わっている研究員の皆様には，貴重なご意見を頂きました．ここに記して感謝の意を表します．

文　献

[1]Whiteside, R., Pancerella, C.M. and Klevgard, P., “A CORBA-Based Manufacturing Environment”, Proc. Hawaii International Conference on System Science, 1997

[2]松家英雄，“機械工場におけるオープン化”，精密工学会誌，Vol.63，No.5，pp.633-638，1997

[3]楠　和浩ら，“分散生産ライン管理・保守システムのオブジェクト指向技術に基づく設計と実装”，情報処理学会論文誌，Vol.40，No.4，pp.1891-1900，1999

[4]Sun Microsystems Computer Corporation, “Java™ Language Specification”, Second Edition, 1996

[5] HORB home page, “http://ring.etl.go.jp/openlab/horb-j/”

[6]日本エクセロン（株）,”OBJECTSTORE PSE/PSE Pro for JAVA API ユーザガイド リリース 3.0”，1998

[7]石塚圭樹，“オブジェクト指向データベース”，アスキー，1996

[8] HDBENCH home page, “http://www.lares.dti.ne.jp/~ep82kazu/”（2000 年 3 月 24 日現在）

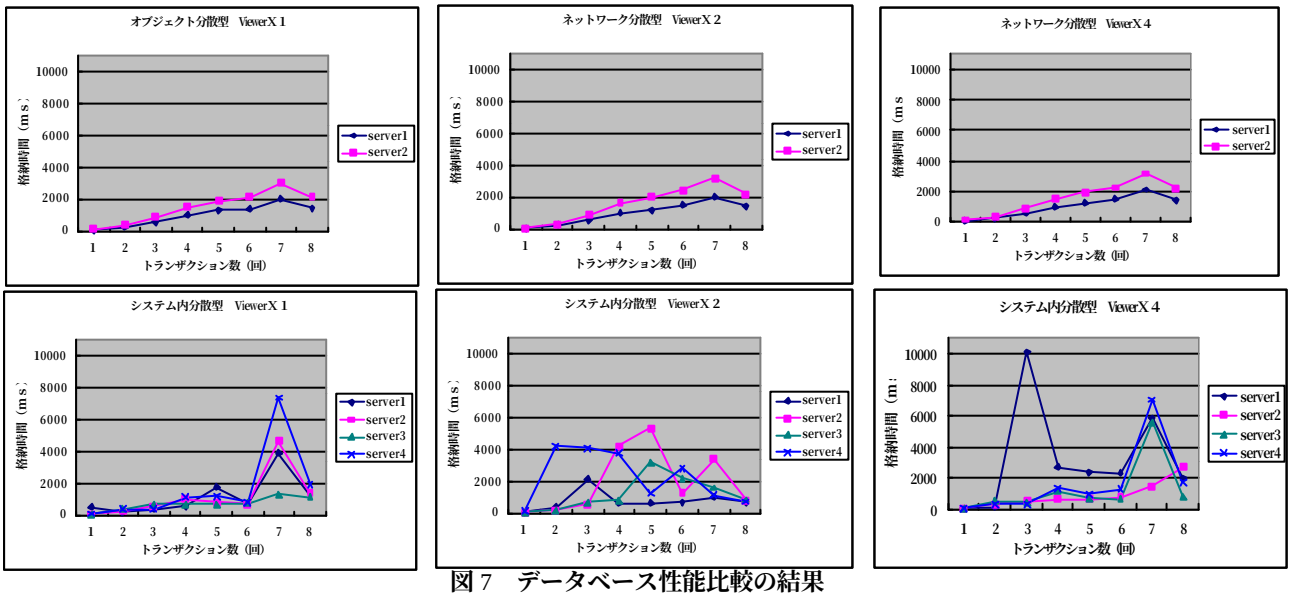


図 7　データベース性能比較の結果

Fig.7 The results of the database performance test

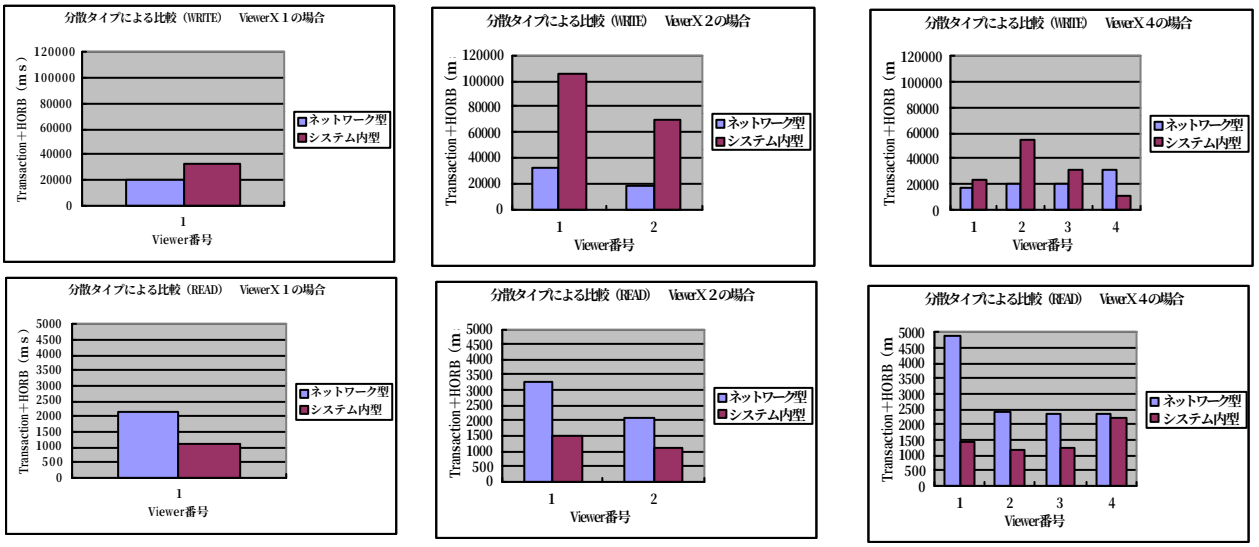


図 8　ORB を含んだ場合の結果

Fig.8 The results of the performance test (includes ORB)