

## 分散データベース環境におけるシステム開発

3V-6

松田 善之  
株式会社 東芝

### 1. はじめに

近年のデータベース（DB）は、オープン化、ダウンサイジング化の流れから広域で大規模のデータを扱うシステムへのニーズが増え、分散DBを取り入れる事への要求が高くなっている。この様な状況に対して、本論文は分散DBを設計する手法を検討、考察した結果を述べるものである。

### 2. 従来の分散DBの実装

従来のDB設計は、ホストやサーバでデータを集中して保持する一箇所集中型のモデルが多かった。しかし、近年ではデータの流用性、トランザクションの分散化、ネットワーク速度によるスループットの劣化への不満、といった要求が多くなってきた。これらの要求を従来のDB設計で実現すると、以下の様に

(1) DB ↔ 一般ファイルデータ変換機能

(2) ファイル転送機能

という2つの機能の組み合わせが一般的だが、

① DBMSの違いを考慮したデータ変換機能の実現

② 通信インターフェースの違いを考慮したファイル転送機能実現

③ データの二重持ちの為におきる双方のデータの不整合解消

④ データの同期をとるタイミング

⑤ データの差分の抽出・反映

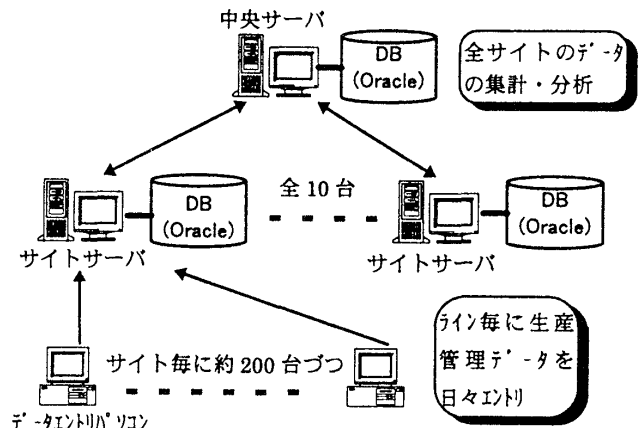
等が非常に困難であり、開発者、保守員にとって非常に扱いづらいシステムとなる事が多かった。

### 3. Oracleの分散DB機能<sup>\*1</sup>

弊社では、複数サイトの生産ラインを管理する必要があるユーザ向けに、生産ライン管理システムを開発した。このシステムは、全サイト（10サイト）と中央とで計11台サーバを配置し、サイト毎のラインの生産管理データをそれぞれのサイトに配置されたサーバにエントリし、中央サーバにてエントリデータの集計・分析を行うシステムである。その為、分散DBとしての特徴が強く、このシステムの分散DBをOracleが実装し

た分散DB機能により実現した。

本システムで検討した分散DB機能の特徴と欠点は、以下のとおりである。



#### (1) DBLINK

【特徴】プロトコルを意識せず他サーバのテーブルへのアクセスを実現

【欠点】アクセス速度はネットワーク性能に依存

#### (2) ReadOnlySnapshot

【特徴】マスターサイトとスナップショットサイトを配置し、マスターデータをスナップショットにコピーする。

【欠点】スナップショットサイトでの更新不可

#### (3) MultiMasterReplication

【特徴】同一テーブルを複数サーバに配置しデータの同期をとる。

【欠点】サーバ数nに対してネットワーク上のトランザクション量が  $Order(n^2)$  で増加

### 4. 生産ライン管理システムへの適用

本システムには、以下の特徴がある。

(1) サイト間のLANは低速。

(2) サイトサーバで利用するデータは自サイトのデータが大半。

(3) 中央サーバでは全サイトのデータを利用する。

これらの特徴を考慮せず、論理設計されたテーブルの物理配置を検討すると、全ての共有するテ

ープルを全サーバに MultiMasterReplication で配置する事が理想である。この時中央サーバで送受信するデータ量を算出すると、1日の更新トランザクション数（中央：100万件、サイト：10万件）より5.6 Gbyte/日となり、同期処理に必要な時間が24時間を超過する予測となる為運用できないと判断した。また、全てのサイトサーバに対して中央サーバと同等の磁気ディスクを装備する必要があり、サイトサーバがオーバスペックとなる。

次に、DBLINK を多用するDB設計を検討する場合、(1)より性能問題によりシステムが要求性能を満たさない事が予想できる。

そこで、以下の様な構成とした。

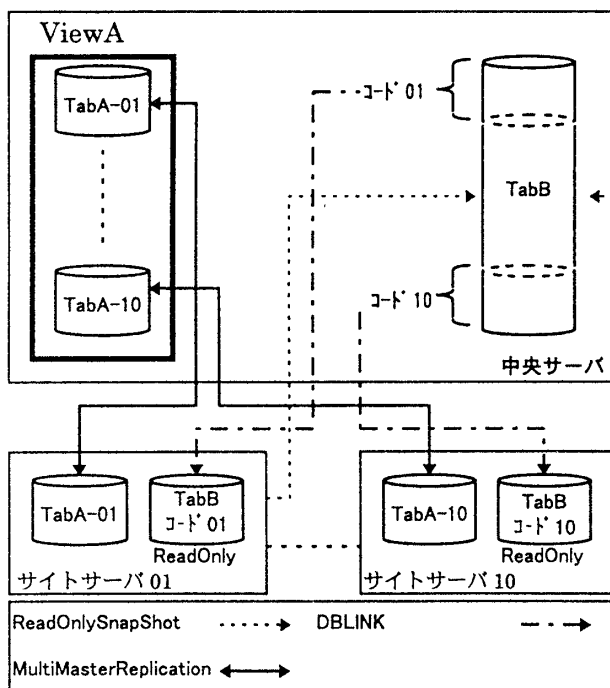


図2 DB設計例

### ① MultiMasterReplication と View の併用

TabA を全サイトのデータを持つテーブルとして1つにまとめず、サイト毎にテーブルを10分割する。そして10分割されたテーブル毎に MultiMasterReplication の定義を行う事により、サイトサーバ、中央サーバ間にサイト毎の更新データのトランザクションを流す事とする。

また、中央サーバでは10分割されたテーブルを UNION で連結した PartitionView を作成し、TabA-01~10 があたかも1つである様に参照可能とする。ただし、View を利用している為、ViewA に対する更新は不可能であり、更新時は、個々のテーブルを指定する必要がある。

### ② SnapShot の分割化と DBLINK の併用

TabB はマスタ系のデータであるが、サイトサーバでは参照のトランザクションがほとんどである為、ReadOnlySnapShot を採用した。また LAN 上のトラフィックを下げる為に、(2)の特徴よりサイトサーバ毎に必要なデータのみ、つまり自サイトのサイトコードで絞り込んだデータのみ TabB コード 01~10 の様に SnapShot として作成した。また、頻度の少ないマスタ系のデータのメンテナンスに関しては、サイトからでも処理可能である様に、DBLINK によりメンテナンスを可能にした。

### 5. 本設計の効果

本設計により、中央サーバで送受信するデータ量を算出すると、1 Gbyte/日となる。中央サーバでのテーブルの同期処理に要する時間は6時間/日と計測されており、当初の予想通り運用可能なDB設計ができた事が確認された。

### 6. 分散DB機能の課題

以下に分散DB機能の課題を述べる。

#### (1) ネットワーク速度への依存度の高さ

MultiMasterReplication、ReadOnlySnapShot で扱えるデータ量はネットワーク速度に依存。

#### (2) 競合解消

MultiMasterReplication において競合(複数サーバで同一データに対し異なる更新処理を実行した場合)が発生すると、解消する方法をロジックで記述する事が困難。

#### (3) 分散DB機能の標準化

分散DB機能が標準化されてない為、DBMS が異なると実装不可能。

### 7. まとめ

性能とコストの関係を考える場合、磁気ディスク、メモリ、プロセッサが安価で高性能である事に対して、高速なネットワーク設備を持つには非常にコストがかかる。よって、サーバに格納できるデータ量や操作可能なデータ量が大幅に増加しても、分散DB環境にこれらのデータを格納する為の十分なネットワーク速度が確保できない状況が起きる事が予想できる。その為、分散DB環境を構築する技術は重要であり、課題の克服が必要である。

### 8. 参考文献

- [1] Oracle7 Server 分散システム Vol. 1: 分散データ (Oracle 社)
- [2] Oracle7 Server 分散システム Vol. 2: レプリカデータ (Oracle 社)

※ 1 Oracle は Oracle 社の商標。