

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6200377号  
(P6200377)

(45) 発行日 平成29年9月20日(2017.9.20)

(24) 登録日 平成29年9月1日(2017.9.1)

(51) Int.Cl. F I  
**G 0 6 F 12/00 (2006.01)**  
 G 0 6 F 12/00 5 1 3 D  
 G 0 6 F 12/00 5 1 4 E

請求項の数 6 (全 21 頁)

(21) 出願番号	特願2014-110806 (P2014-110806)	(73) 特許権者	000004226 日本電信電話株式会社 東京都千代田区大手町一丁目5番1号
(22) 出願日	平成26年5月29日 (2014.5.29)	(74) 代理人	110001807 特許業務法人磯野国際特許商標事務所
(65) 公開番号	特開2015-225563 (P2015-225563A)	(72) 発明者	武 直樹 東京都千代田区大手町一丁目5番1号 日 本電信電話株式会社内
(43) 公開日	平成27年12月14日 (2015.12.14)	(72) 発明者	西尾 学 東京都千代田区大手町一丁目5番1号 日 本電信電話株式会社内
審査請求日	平成28年7月22日 (2016.7.22)	審査官	桜井 茂行

最終頁に続く

(54) 【発明の名称】 仮想DBシステム、および、仮想DBシステムの情報処理方法

(57) 【特許請求の範囲】

【請求項1】

複数のデータソースを統合して、複数の前記データソースそれぞれとクライアント A P (Application) との間の情報を処理する仮想 D B (DataBase) サーバと、前記仮想 D B サーバに接続され、複数の前記データソースそれぞれとの間で情報の送受信を行い、複数の前記データソースそれぞれに対応付けて設けられるアダプタと、を備える仮想 D B システムであって、

複数の前記データソースには、前記データソース自身のイベント発生時に、イベント通知を発信する機能を備えた前記データソースが含まれており、

前記仮想 D B サーバは、前記クライアント A P から監視対象となる前記データソースの識別情報と監視条件とを含む監視対象情報が記載された参照クエリを受信し、前記参照クエリを実行することにより、前記監視対象情報で示される特定のデータソースに対応するアダプタに、前記監視対象情報を送信するクエリ実行部を備え、

前記アダプタは、受信した前記監視対象情報で示される前記特定のデータソースを監視対象として登録し、前記特定のデータソースから前記イベント通知を受け取ると、前記仮想 D B サーバに前記監視条件に合致する当該イベント通知の情報を送信するイベント通知処理部を備え、

前記仮想 D B サーバのクエリ実行部は、

前記参照クエリを実行することにより、前記特定のデータソースに対応するアダプタから前記監視条件に合致する前記イベント通知の情報を取得しておき、前記参照クエリの実

10

20

行を終了し、

前記参照クエリのデータ読み出し処理であるフェッチ時に、前記クライアント A P へ前記取得したイベント通知の情報を送信すること

を特徴とする仮想 D B システム。

【請求項 2】

複数のデータソースを統合して、複数の前記データソースそれぞれとクライアント A P との間の情報を処理する仮想 D B サーバと、前記仮想 D B サーバに接続され、複数の前記データソースそれぞれとの間で情報の送受信を行い、複数の前記データソースそれぞれに対応付けて設けられるアダプタと、を備える仮想 D B システムであって、

複数の前記データソースには、前記データソース自身のイベント発生時に、イベント通知を発信する機能を備えた前記データソースが含まれており、

前記仮想 D B サーバは、前記クライアント A P から監視対象となる前記データソースの識別情報と監視条件とを含む監視対象情報が記載された参照クエリを受信し、前記参照クエリを実行することにより、前記監視対象情報で示される特定のデータソースに対応するアダプタに、前記監視対象情報を送信するクエリ実行部を備え、

前記アダプタは、受信した前記監視対象情報で示される前記特定のデータソースを監視対象として登録し、前記特定のデータソースから前記イベント通知を受け取るイベント通知処理部を備え、

前記アダプタの前記イベント通知処理部は、通知蓄積部を備えており、

前記特定のデータソースから前記イベント通知を受け取ると、前記監視条件に合致する当該イベント通知を前記通知蓄積部に記憶しておき、

前記仮想 D B サーバのクエリ実行部は、

前記参照クエリを実行することにより、前記特定のデータソースに対応するアダプタに、前記特定のデータソースを監視対象として登録すると、前記参照クエリの実行を終了し、

前記参照クエリのデータ読み出し処理であるフェッチ時に、前記アダプタの通知蓄積部に問い合わせることにより、前記イベント通知の情報を取得し、前記クライアント A P へ前記取得したイベント通知の情報を送信すること

を特徴とする仮想 D B システム。

【請求項 3】

前記仮想 D B サーバのクエリ実行部は、前記フェッチ時に、前記アダプタの通知蓄積部に問い合わせた結果、当該通知蓄積部が前記イベント通知を記憶していなかった場合に、前記特定のデータソースに対応するアダプタが前記イベント通知を受け取り、当該イベント通知が前記通知蓄積部に記憶されるまで待機すること

を特徴とする請求項 2 に記載の仮想 D B システム。

【請求項 4】

複数のデータソースを統合して、複数の前記データソースそれぞれとクライアント A P との間の情報を処理する仮想 D B サーバと、前記仮想 D B サーバに接続され、複数の前記データソースそれぞれとの間で情報の送受信を行い、複数の前記データソースそれぞれに対応付けて設けられるアダプタと、を備える仮想 D B システムの情報処理方法であって、

複数の前記データソースには、前記データソース自身のイベント発生時に、イベント通知を発信する機能を備えた前記データソースが含まれており、

前記仮想 D B サーバは、前記クライアント A P から監視対象となる前記データソースの識別情報と監視条件とを含む監視対象情報が記載された参照クエリを受信し、前記参照クエリを実行することにより、前記監視対象情報で示される特定のデータソースに対応するアダプタに、前記監視対象情報を送信し、

前記アダプタは、受信した前記監視対象情報で示される前記特定のデータソースを監視対象として登録し、前記特定のデータソースから前記イベント通知を受け取ると、前記仮想 D B サーバに前記監視条件に合致する当該イベント通知の情報を送信し、

前記仮想 D B サーバは、

10

20

30

40

50

前記参照クエリを実行することにより、前記特定のデータソースに対応するアダプタから前記監視条件に合致する前記イベント通知の情報を取得しておき、前記参照クエリの実行を終了し、

前記参照クエリのデータ読み出し処理であるフェッチ時に、前記クライアント A P へ前記取得したイベント通知の情報を送信すること

を特徴とする仮想 D B システムの情報処理方法。

【請求項 5】

複数のデータソースを統合して、複数の前記データソースそれぞれとクライアント A P との間の情報を処理する仮想 D B サーバと、前記仮想 D B サーバに接続され、複数の前記データソースそれぞれとの間で情報の送受信を行い、複数の前記データソースそれぞれに

10

対応付けて設けられるアダプタと、を備える仮想 D B システムの情報処理方法であって、

複数の前記データソースには、前記データソース自身のイベント発生時に、イベント通知を発信する機能を備えた前記データソースが含まれており、

前記仮想 D B サーバは、前記クライアント A P から監視対象となる前記データソースの識別情報と監視条件とを含む監視対象情報が記載された参照クエリを受信し、前記参照クエリを実行することにより、前記監視対象情報で示される特定のデータソースに対応するアダプタに、前記監視対象情報を送信し、

前記アダプタは、通知蓄積部を備えており、

受信した前記監視対象情報で示される前記特定のデータソースを監視対象として登録し、前記特定のデータソースから前記イベント通知を受け取ると、前記監視条件に合致する

20

当該イベント通知を前記通知蓄積部に記憶しておき、

前記仮想 D B サーバは、

前記参照クエリを実行することにより、前記特定のデータソースに対応するアダプタに、前記特定のデータソースを監視対象として登録すると、前記参照クエリの実行を終了し、

前記参照クエリのデータ読み出し処理であるフェッチ時に、前記アダプタの通知蓄積部に問い合わせることにより、前記イベント通知の情報を取得し、前記クライアント A P へ前記取得したイベント通知の情報を送信すること

を特徴とする仮想 D B システムの情報処理方法。

【請求項 6】

30

前記仮想 D B サーバは、前記フェッチ時に、前記アダプタの通知蓄積部に問い合わせた結果、当該通知蓄積部が前記イベント通知を記憶していなかった場合に、前記特定のデータソースに対応するアダプタが前記イベント通知を受け取り、当該イベント通知が前記通知蓄積部に記憶されるまで待機すること

を特徴とする請求項 5 に記載の仮想 D B システムの情報処理方法。

【発明の詳細な説明】

【技術分野】

【0001】

異なる種類の複数のデータソースを単一のデータソースとしてクライアント A P (Application) に見せる仮想 D B (DataBase) システム、および、仮想 D B システムの情報処理方法に関する。

40

【背景技術】

【0002】

昨今のネットワークの高度化やサービスライフスタイルの短命化に伴い、その運用を支援する O S S (Operation Support System) は、サービス毎にその提供時期に合わせて個別にかつ短期間で開発・導入されている。この個別に開発された O S S が乱立することで、通信事業者の O S S の構成は複雑になり、O S S の開発・維持コストが増加する一因となっている。

【0003】

図 9 ( a ) は、個別に開発された O S S とその O S S に接続された N E (Network Elem

50

ent) 5 とを含むシステムを示す図である。O S S 毎に、対応する各クライアント A P 2 (以下、単に「A P」と記載する場合がある。)や D B 3 が開発され、図 9 ( a ) に示すように、その各 A P 2 と N E 5 とがそれぞれ接続される。

このようなシステム構成において、個別に開発された O S S の開発・維持コストが増加するのは、以下に示す、( 1 ) A P 間連携の複雑化、( 2 ) 個別 D B 内のデータ重複、( 3 ) N E アクセスのための機能重複、が主な原因となる。

【 0 0 0 4 】

( 1 ) A P 間連携の複雑化

O S S の A P 2 同士が個別に 1 対 1 で連携する場合、全体として連携がメッシュ構成となる。仮に n 個の A P 2 がメッシュ構成で連携を行った場合、n の 2 乗オーダの I F (インタフェース) 開発が必要になる。

【 0 0 0 5 】

( 2 ) 個別 D B 内のデータ重複

異なる O S S で同じデータが管理されている場合、情報の一貫性を維持するためのデータ同期等の機能を要する。

【 0 0 0 6 】

( 3 ) N E アクセスのための機能重複

複数の O S S 上の A P 2 が個別に N E アクセスの I F を開発した結果、同等または類似のデータを参照・更新する機能の重複が生じる場合がある。

【 0 0 0 7 】

このような問題を解決するため、異なる種類の複数のデータソース ( D B M S ( Database Management System ) やファイルシステム等 ) をクライアント A P に対して 1 つのデータソースとして統合して見せる技術として仮想 D B システム ( 以下、単に「仮想 D B」とよぶ場合がある。 ) がある ( 例えば、非特許文献 1 参照 ) 。図 9 ( b ) に示す従来の仮想 D B では、複数の D B 3 を仮想 D B システム 1 として統合し、各 A P 2 は仮想 D B システム 1 に対してアクセスすることで、各 D B 3 に関する情報処理を実行することができる。

【 0 0 0 8 】

この図 9 ( b ) に示すような仮想 D B システム 1 を構築し、各 D B 3 を論理的に統合することで、「( 2 ) 個別 D B 内のデータ重複」の問題を解決し、同時に、データが共有されることにより、データ流通を目的とした A P 間連携が必要なくなるため、「( 1 ) A P 連携の複雑化」の問題も解決することができる。

【 先行技術文献 】

【 非特許文献 】

【 0 0 0 9 】

【 非特許文献 1 】 Red Hat、[ online ]、[ 平成 26 年 5 月 16 日 検索 ]、インターネット < U R L : <http://jp.redhat.com/products/jbossenterprise middleware/data-services/> >

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 1 0 】

しかしながら、従来の仮想 D B が想定しているデータソースは、D B やファイル等の静的なものであり、自身が記憶する情報に変更があった場合に自ら ( つまり「動的に」 ) 情報を発信するようなデータソース ( 例えば、N E や、他の A P 等 ) からの情報を処理する機構を備えていなかった。N E 等が動的に発信する情報としては、例えば、故障等の障害の発生を知らせるアラームや、N E の管理情報 ( C P U ( Central Processing Unit ) 使用率やメモリ使用率等 ) のような時間経過により変化する情報についての発信情報等 ( 以下、これらを総称して「イベント通知」と称する。 ) があげられる。従来の仮想 D B は、このような動的に情報を発信するデータソースには対応していないため、「( 3 ) N E アクセスのための機能重複」の問題を解決し N E も含めて仮想 D B に統合する、つまり、N E に対する操作や、N E から発信される情報を処理する操作も含めてすべて仮想 D B 中のデータに対する処理として扱う、ことについては実現していなかった。

10

20

30

40

50

## 【 0 0 1 1 】

このような背景に鑑みて本発明がなされたのであり、本発明は、データソース側から発信される情報（イベント通知）も含めて仮想DB内で情報を一元的に処理することができる、仮想DBシステム、および、仮想DBシステムの情報処理方法を提供することを課題とする。

つまり、従来のDBやファイル等の静的なデータソースに加えて、イベント通知を送信するような動的なデータソース（NE等）を含めて仮想DBに統合し情報処理することができる、仮想DBシステム、および、仮想DBシステムの情報処理方法を提供することを課題とする。

## 【課題を解決するための手段】

10

## 【 0 0 1 2 】

前記した課題を解決するため、請求項1に記載の発明は、複数のデータソースを統合して、複数の前記データソースそれぞれとクライアントAP（Application）との間の情報を処理する仮想DB（DataBase）サーバと、前記仮想DBサーバに接続され、複数の前記データソースそれぞれとの間で情報の送受信を行い、複数の前記データソースそれぞれに対応付けて設けられるアダプタと、を備える仮想DBシステムであって、複数の前記データソースには、前記データソース自身のイベント発生時に、イベント通知を発信する機能を備えた前記データソースが含まれており、前記仮想DBサーバが、前記クライアントAPから監視対象となる前記データソースの識別情報と監視条件とを含む監視対象情報が記載された参照クエリを受信し、前記参照クエリを実行することにより、前記監視対象情報で示される特定のデータソースに対応するアダプタに、前記監視対象情報を送信するクエリ実行部を備え、前記アダプタが、受信した前記監視対象情報で示される前記特定のデータソースを監視対象として登録し、前記特定のデータソースから前記イベント通知を受け取ると、前記仮想DBサーバに前記監視条件に合致する当該イベント通知の情報を送信するイベント通知処理部を備え、前記仮想DBサーバのクエリ実行部が、前記参照クエリを実行することにより、前記特定のデータソースに対応するアダプタから前記監視条件に合致する前記イベント通知の情報を取得しておき、前記参照クエリの実行を終了し、前記参照クエリのデータ読み出し処理であるフェッチ時に、前記クライアントAPへ前記取得したイベント通知の情報を送信すること、を特徴とする仮想DBシステムとした。

20

## 【 0 0 1 3 】

30

また、請求項4に記載の発明は、複数のデータソースを統合して、複数の前記データソースそれぞれとクライアントAPとの間の情報を処理する仮想DBサーバと、前記仮想DBサーバに接続され、複数の前記データソースそれぞれとの間で情報の送受信を行い、複数の前記データソースそれぞれに対応付けて設けられるアダプタと、を備える仮想DBシステムの情報処理方法であって、複数の前記データソースには、前記データソース自身のイベント発生時に、イベント通知を発信する機能を備えた前記データソースが含まれており、前記仮想DBサーバが、前記クライアントAPから監視対象となる前記データソースの識別情報と監視条件とを含む監視対象情報が記載された参照クエリを受信し、前記参照クエリを実行することにより、前記監視対象情報で示される特定のデータソースに対応するアダプタに、前記監視対象情報を送信し、前記アダプタが、受信した前記監視対象情報で示される前記特定のデータソースを監視対象として登録し、前記特定のデータソースから前記イベント通知を受け取ると、前記仮想DBサーバに前記監視条件に合致する当該イベント通知の情報を送信し、前記仮想DBサーバが、前記参照クエリを実行することにより、前記特定のデータソースに対応するアダプタから前記監視条件に合致する前記イベント通知の情報を取得しておき、前記参照クエリの実行を終了し、前記参照クエリのデータ読み出し処理であるフェッチ時に、前記クライアントAPへ前記取得したイベント通知の情報を送信することを特徴とする仮想DBシステムの情報処理方法とした。

40

## 【 0 0 1 4 】

このようにすることで、仮想DBシステムは、データソースから発信されたイベント通知をアダプタが受け取り、クライアントAPからの参照クエリに対する応答として、仮想

50

DBサーバがそのイベント通知の情報をクライアントAPに送信することができる。

よって、従来のDBやファイル等の静的なデータソースに加えて、イベント通知を送信するような動的なデータソース(NE5等)を含めて仮想DBに統合し情報処理することができる。

【0017】

また、仮想DBシステムは、データソースから発信されたイベント通知を、クエリ実行時に仮想DBサーバが取得し、クライアントAPからの参照クエリに対する応答として、そのイベント通知の情報をクライアントAPに送信することができる。

【0018】

請求項2に記載の発明は、複数のデータソースを統合して、複数の前記データソースそれぞれとクライアントAPとの間の情報を処理する仮想DBサーバと、前記仮想DBサーバに接続され、複数の前記データソースそれぞれとの間で情報の送受信を行い、複数の前記データソースそれぞれに対応付けて設けられるアダプタと、を備える仮想DBシステムであって、複数の前記データソースには、前記データソース自身のイベント発生時に、イベント通知を発信する機能を備えた前記データソースが含まれており、前記仮想DBサーバが、前記クライアントAPから監視対象となる前記データソースの識別情報と監視条件とを含む監視対象情報が記載された参照クエリを受信し、前記参照クエリを実行することにより、前記監視対象情報で示される特定のデータソースに対応するアダプタに、前記監視対象情報を送信するクエリ実行部を備え、前記アダプタが、受信した前記監視対象情報で示される前記特定のデータソースを監視対象として登録し、前記特定のデータソースから前記イベント通知を受け取るイベント通知処理部を備え、前記アダプタの前記イベント通知処理部が、通知蓄積部を備えており、前記特定のデータソースから前記イベント通知を受け取ると、前記監視条件に合致する当該イベント通知を前記通知蓄積部に記憶しておき、前記仮想DBサーバのクエリ実行部が、前記参照クエリを実行することにより、前記特定のデータソースに対応するアダプタに、前記特定のデータソースを監視対象として登録すると、前記参照クエリの実行を終了し、前記参照クエリのデータ読み出し処理であるフェッチ時に、前記アダプタの通知蓄積部に問い合わせることにより、前記イベント通知の情報を取得し、前記クライアントAPへ前記取得したイベント通知の情報を送信することを特徴とする仮想DBシステムとした。

【0019】

また、請求項5に記載の発明は、複数のデータソースを統合して、複数の前記データソースそれぞれとクライアントAPとの間の情報を処理する仮想DBサーバと、前記仮想DBサーバに接続され、複数の前記データソースそれぞれとの間で情報の送受信を行い、複数の前記データソースそれぞれに対応付けて設けられるアダプタと、を備える仮想DBシステムの情報処理方法であって、複数の前記データソースには、前記データソース自身のイベント発生時に、イベント通知を発信する機能を備えた前記データソースが含まれており、前記仮想DBサーバが、前記クライアントAPから監視対象となる前記データソースの識別情報と監視条件とを含む監視対象情報が記載された参照クエリを受信し、前記参照クエリを実行することにより、前記監視対象情報で示される特定のデータソースに対応するアダプタに、前記監視対象情報を送信し、前記アダプタが、通知蓄積部を備えており、受信した前記監視対象情報で示される前記特定のデータソースを監視対象として登録し、前記特定のデータソースから前記イベント通知を受け取ると、前記監視条件に合致する当該イベント通知を前記通知蓄積部に記憶しておき、前記仮想DBサーバが、前記参照クエリを実行することにより、前記特定のデータソースに対応するアダプタに、前記特定のデータソースを監視対象として登録すると、前記参照クエリの実行を終了し、前記参照クエリのデータ読み出し処理であるフェッチ時に、前記アダプタの通知蓄積部に問い合わせることにより、前記イベント通知の情報を取得し、前記クライアントAPへ前記取得したイベント通知の情報を送信することを特徴とする仮想DBシステムの情報処理方法とした。

【0020】

このようにすることで、仮想DBシステムは、データソースから発信されたイベント通

10

20

30

40

50

知をアダプタが受け取り、クライアント A P からの参照クエリに対する応答として、仮想 D B サーバがそのイベント通知の情報をクライアント A P に送信することができる。

よって、従来の D B やファイル等の静的なデータソースに加えて、イベント通知を送信するような動的なデータソース ( N E 5 等 ) を含めて仮想 D B に統合し情報処理することができる。

また、仮想 D B システムは、データソースから発信されたイベント通知の情報を、参照クエリのフェッチ時に仮想 D B サーバがアダプタの通知蓄積部から取得し、そのイベント通知の情報をクライアント A P に送信することができる。

【 0 0 2 1 】

請求項 3 に記載の発明は、前記仮想 D B サーバのクエリ実行部が、前記フェッチ時に、前記アダプタの通知蓄積部に問い合わせた結果、当該通知蓄積部が前記イベント通知を記憶していなかった場合に、前記特定のデータソースに対応するアダプタが前記イベント通知を受け取り、当該イベント通知が前記通知蓄積部に記憶されるまで待機することを特徴とする請求項 2 に記載の仮想 D B システムとした。

10

【 0 0 2 2 】

また、請求項 6 に記載の発明は、前記仮想 D B サーバが、前記フェッチ時に、前記アダプタの通知蓄積部に問い合わせた結果、当該通知蓄積部が前記イベント通知を記憶していなかった場合に、前記特定のデータソースに対応するアダプタが前記イベント通知を受け取り、当該イベント通知が前記通知蓄積部に記憶されるまで待機することを特徴とする請求項 5 に記載の仮想 D B システムの情報処理方法とした。

20

【 0 0 2 3 】

このようにすることで、仮想 D B システムは、アダプタの通知蓄積部にイベント通知が記憶されていない場合においても実行される、無駄となるフェッチの処理、それに伴う問い合わせや応答処理をなくすることができる。よって、仮想 D B サーバおよびクライアント A P の処理負荷を低減することができる。

【発明の効果】

【 0 0 2 4 】

本発明によれば、イベント通知を発信する機能を備えたデータソースも含めて仮想 D B 内で情報を一元的に処理する、仮想 D B システム、および、仮想 D B システムの情報処理方法を提供することができる。

30

【図面の簡単な説明】

【 0 0 2 5 】

【図 1】本実施形態に係る仮想 D B システムの全体構成を説明するための図である。

【図 2】本実施形態に係る仮想 D B システムの構成の概要を説明するための図である。

【図 3】参照クエリによる、D B 内のデータの参照処理を説明するための図である。

【図 4】本発明の第 1 の実施形態に係る仮想 D B システムの構成例を示す機能ブロック図である。

【図 5】本発明の第 1 の実施形態に係る仮想 D B システムの処理の流れを示すフローチャートである。

【図 6】本発明の第 2 の実施形態に係る仮想 D B システムの構成例を示す機能ブロック図である。

40

【図 7】本発明の第 2 の実施形態に係る仮想 D B システムの処理の流れを示すフローチャートである。

【図 8】本発明の第 2 の実施形態の変形例に係る仮想 D B システムの処理の流れを示すフローチャートである。

【図 9】従来の仮想 D B システムによる複数の D B の統合を説明するための図である。

【図 10】比較例の仮想 D B システムの構成例を示す機能ブロック図である。

【図 11】比較例の仮想 D B システムにおいて、イベント通知を処理する手法を例示する図である。

【発明を実施するための形態】

50

## 【0026】

次に、本発明を実施するための形態（以下、「本実施形態」という。）における仮想DBシステム1000等について説明する。

まず、比較例として従来の仮想DBシステム1を説明し、その後、本実施形態に係る仮想DBシステム1000について説明する。

## 【0027】

<比較例の仮想DBシステム>

図10は、比較例の仮想DBシステム1の構成例を示す機能ブロック図である。

比較例の仮想DBシステム1は、クライアントAP（以下、単に「AP」と称する。）2からクエリを受け取り、そのクエリを各データソース用のクエリに変換した上で、各データソース用のアダプタ30を介してそのクエリを出力し各データソースに実行させる。ここでの各データソースは、DBMS（図では「DB3」と記載する。）やファイルシステム（図では、「ファイル4」と記載する。）等であり、イベント通知等を送信する機能を備えたデータソース（例えば、NE）は含んでいない。

この比較例の仮想DBシステム1は、図10に示すように、仮想DBサーバ100と、データソースそれぞれに対応した複数のアダプタ30（30D<sub>1</sub>、30D<sub>2</sub>、30F）とを備えて構成される。そして、この仮想DBサーバ100は、クエリエンジン10およびスキーマ・ルール格納部20を備える。

## 【0028】

クエリエンジン10は、AP2から、例えばSQL（Structured Query Language）で記述されたクエリを受け取り、そのクエリを各データソース用のクエリに変換し、実行計画を立て実行する。このクエリエンジン10は、クエリ解析・書換部11、クエリ最適化部12およびクエリ実行部13を備える。

## 【0029】

クエリ解析・書換部11は、AP2からのクエリを受け取り、そのクエリの字句解析を行う。そして、クエリ解析・書換部11は、解析したクエリに対して、スキーマ・ルール格納部20を参照し、クライアントAP用スキーマからデータソース（DS）用スキーマへのクエリの書き換えを行う。

なお、クライアントAP用スキーマとは、AP2側に見せるためのスキーマであり、AP2が参照可能な形式（例えば、テーブル形式）でデータ構造が格納されるものである。また、データソース（DS）用スキーマとは、データソースそれぞれに対応したデータ構造が格納されるものである。

## 【0030】

クエリ最適化部12は、クエリ解析・書換部11で書き換えられたクエリに対して、最適な実行計画を策定する。

クエリ実行部13は、策定された実行計画を、そのクエリが要求する処理対象のデータソース用のアダプタ30を介して実行する。

## 【0031】

スキーマ・ルール格納部20には、不図示の記憶手段に記憶される情報であり、AP2に見せるクライアントAP用スキーマ（図示省略）、各データソースに対応したデータソース（DS）用スキーマ（図示省略）、このクライアントAP用スキーマとデータソース（DS）用スキーマとを対応付けるマッピングルール（図示省略）等が格納される。

## 【0032】

アダプタ30は、各データソースに対応して設けられる。例えば、アダプタ30（30D<sub>1</sub>）は、DB「1」用のアダプタであり、アダプタ30（30D<sub>2</sub>）は、DB「2」用のアダプタであり、アダプタ30（30F）は、ファイル（ファイルシステム）用のアダプタである。そして、各アダプタ30は、クエリ実行部13が実行したクエリを各データソースにおけるクエリへ変換する。

## 【0033】

比較例の仮想DBシステム1は、このように、データソースそれぞれに対応したアダプ

10

20

30

40

50

タ30を備え、仮想DBシステム1に対するクエリを各データソースに対するクエリに変換することにより、各データソース(DBMSやファイルシステム)の仮想的な統合を実現している。

#### 【0034】

しかしながら、前記したように、比較例の仮想DBシステム1では、データソース(DS)側で発生したイベント通知を検知して処理を実行する機能を備えていない。そのため、仮に比較例の仮想DBシステム1を用いて、データソース(DS)側からのイベント通知を処理しようとする、図11に示すように、新たに外部のAP(外部AP)2(2a)を設けて、AP側でイベント通知を受け取り、その通知を解析し、仮想DBシステム1に投入するためのクエリに変換する必要がある。図11では、外部AP2(2a)に、通知受付機能と、通知-クエリ変換機能と、クエリ投入機能とを備えた例を示している。NE5(5N)からのイベント通知を、外部AP2の通知受付機能が受け付け、通知-クエリ変換機能が、イベント通知を解析し仮想DBシステム1に対応したクエリに変換した上で、クエリ投入機能が仮想DBシステム1に対して、クエリを送信する。

10

#### 【0035】

このように、外部AP2(2a)を設けることで、データソース(DS)側で発生したイベント通知を仮想DBシステム1において処理することは可能であるが、AP側でイベント通知をクエリに変換する必要が生じることや、仮想DBシステム1内のクエリエンジン10を経由することになりオーバーヘッドが大きくなるという問題があった。

#### 【0036】

よって、従来のDB3やファイル4等のデータソースに加えて、NE5等のイベント通知を発信するような動的なデータソースを含めて仮想DBに統合し情報処理することは、実現していなかった。

20

#### 【0037】

(本実施形態)

次に、本実施形態に係る仮想DBシステム1000等について説明する。

#### 【0038】

<概要>

上記において説明したように、従来の仮想DBが想定しているデータソースは、DB3やファイル4等の静的なものである。よって、従来の仮想DBは、自らイベント通知を発信するようなデータソースからの情報を処理する機能を備えていなかった。

30

#### 【0039】

これに対し、本実施形態に係る仮想DBシステム1000は、図1に示すように、データソースとして、DB3やファイル4に加え、自らイベント通知を発信する機能を備えた装置である、NE5等も含めて統合し情報を一元的に扱う。つまり、例えば、管理対象装置としてのNE5との間のアクセスや、他のAP2との連携等を含めて、仮想DBシステム1000を通して行う。

#### 【0040】

そのため、仮想DBシステム1000は、各データソースに対応したアダプタ30を備える。特に、本実施形態では、図2に示すように、自らイベント通知を発信する機能を備えた装置である、NE5等に対応したアダプタ30(図2ではアダプタ30N)を備える。

40

そして、このNE用のアダプタ30(30N)が、比較例における仮想DBシステム1(図10)のアダプタ30にはない、データソース側からのイベント通知を処理するためのイベント通知処理部300を備えることを特徴とする。このイベント通知処理部300は、データソースから発信されたイベント通知を受け取り、AP2からの参照クエリに対する応答として、仮想DBサーバ100を介して、そのイベント通知の情報を送信する処理を実行する。

つまり、従来のアダプタ30の機能を拡張し、イベント通知処理部300を備えるようにすることにより、AP2からの参照クエリに対する応答の形で、データソースからのイ

50

イベント通知の情報を、仮想DBシステム1000からAP2側に送信する。

#### 【0041】

このようにすることで、本実施形態に係る仮想DBシステム1000、および、仮想DBシステムの情報処理方法によれば、従来のDBやファイル等の静的なデータソースに加えて、イベント通知を送信するような動的なデータソース(NE5等)を含めて仮想DBに統合し情報処理することができる。

また、従来のAP2を極力変更することなく、データソースからのイベント通知を従来のDBアクセスと同様に扱うことができる。さらに、従来の仮想DBサーバ100の改編を行うことなく、データソースからのイベント通知を扱うことができる。なお、詳細は後記する。

#### 【0042】

本実施形態においては、上記のように、仮想DBシステム1000が、アダプタ30内にイベント通知処理部300を備えることにより、AP2からの参照クエリに対する応答の形で、データソースからのイベント通知の情報をAP2側に送信することを特徴とするものである。通常、DB3内のデータ処理を参照するためには、AP2からDB3に対して、次のような操作が行われる。

図3に示すように、まず、AP2は、DB3に対して、DB参照クエリを発行する(ステップS1)。図3では、AP2が、DB3に記憶されている「table1」からデータを取得する例を示している(SELECT \* FROM table1;)。ステップS1により、DB3において条件に適合するレコードの一覧が作成される。

次に、AP2が、DB3からレコード(データ:図3では、col1とcol2のデータ)を順次読み出す処理であるフェッチ(各図において「fetch」と記載する。)を実行する。このフェッチにより、AP2は、DB3から各データを取得する(ステップS2)。

なお、図3の符号Pには、AP2がクエリを実行し、フェッチによりデータを取得するためのプログラム例を示している。

#### 【0043】

本実施形態では、AP2から仮想DBシステム1000に対する参照クエリにより、(最初に)監視対象装置(例えば、監視対象となるNE5)と監視条件とをアダプタ30内のイベント通知処理部300に登録しておく。そして、アダプタ30が、登録した監視対象装置と監視条件に適合するイベント通知を受信した場合に、AP2からの参照クエリに対する応答として、そのイベント通知の情報をAP2へ送信する処理を行う。

#### 【0044】

ここで、本実施形態に係る仮想DBシステム1000において、仮想DBサーバ100がアダプタ30から、イベント通知の情報(イベント情報)を取得するタイミングとしては、以下の2つのパターンがある。

(パターン1)クエリ実行時にイベント情報を取得する方式

(パターン2)フェッチ時にイベント情報を取得する方式

以下において、この(パターン1)のクエリ実行時に仮想DBサーバ100がイベント情報を取得する方式を、本発明の第1の実施形態として説明し、(パターン2)のフェッチ時に仮想DBサーバ100がイベント情報を取得する方式を、本発明の第2の実施形態として説明する。

#### 【0045】

<第1の実施形態>

まず、仮想DBシステム1000のアダプタ30が備えるイベント通知処理部300を介して、(パターン1)クエリ実行時に仮想DBサーバ100がイベント情報を取得する方式について具体的に説明する。

なお、ここで「クエリ実行時」とは、仮想DBサーバ100がAP2から参照クエリを受信してから、その参照クエリを解析して実行し、後記するクエリ実行終了メッセージをAP2に送信するまでの期間を意味し、その後のフェッチ処理の実行期間を含まない。

#### 【0046】

図4は、本発明の第1の実施形態に係る仮想DBシステム1000の構成例を示す機能ブロック図である。

図4に示した仮想DBシステム1000は、図10に示した比較例の仮想DBシステム1と比べ、アダプタ30内に、イベント通知処理部300を備える点が異なる。なお、図10に示した仮想DBシステム1の仮想DBサーバ100と、本発明の第1の実施形態に係る仮想DBシステム1000の仮想DBサーバ100とは、同一の機能を備えるものであるため、同一の符号と名称を付し、詳細な説明は省略する。また、図4においては、仮想DBシステム1000にNE5が1台のみ接続された例を示しているが、図1に示したように、複数のDB3やファイルシステム4、他のNE5やAP2がデータソースとして接続されていてもよい。

10

#### 【0047】

本発明の第1の実施形態に係るアダプタ30は、クエリエンジン10のクエリ実行部13が実行したクエリを対応するデータソース（ここでは、NE5）におけるクエリやそのデータソースに適合するプロトコルに基づく操作（コマンド）等に変換する処理を行う。そして、このアダプタ30は、さらに、図4に示すイベント通知処理部300の機能により、データソースから発信されたイベント通知を受け取り、AP2からの参照クエリに対する応答として、仮想DBサーバ100を介して、そのイベント通知を送信する。より詳細には、イベント通知処理部300は、クエリ実行時に、監視対象装置（ここでは、NE5）と監視条件とを登録しておき、監視対象装置から監視条件に合致するイベント通知を受け付けた場合に、そのイベント通知の情報（イベント情報）を仮想DBサーバ100に送信し記憶させておく。つまり、仮想DBシステム1000は、クエリ実行時にイベント通知の情報（イベント情報）をアダプタ30のイベント通知処理部300から仮想DBサーバ100が取得し、記憶しておくようにする。そして、AP2は、フェッチにより仮想DBサーバ100に記憶されたイベント情報を取得する。

20

このイベント通知処理部300は、図4に示すように、仮想DB側IF部310と、通知受付部320とを備える。

#### 【0048】

仮想DB側IF部310は、仮想DBサーバ100のクエリ実行部13から、クエリ形式で取得した監視対象装置（ここでは、NE5）および監視条件の情報（後記する「監視対象情報」）を解釈し、通知受付部320に登録することにより、監視を開始させる。

30

また、仮想DB側IF部310は、通知受付部320が、登録した監視対象装置から監視対象条件に合致するイベント通知（後記する図5においては、「trap」と記載する。）を受け付けた場合に、そのイベント通知（「trap」）を取得して、仮想DBサーバ100にイベント情報として送信する。

#### 【0049】

通知受付部320は、仮想DB側IF部310からの要求により、監視対象装置および監視条件を登録する。そして、通知受付部320は、登録した監視対象装置からイベント通知を受け付けた場合に、登録した監視条件に合致するか否かを判定し、監視条件に合致する場合に、そのイベント通知を仮想DB側IF部310に出力する。

#### 【0050】

40

仮想DBサーバ100（クエリ実行部13）は、アダプタ30の仮想DB側IF部310からイベント情報を受信すると、そのイベント情報を自身の記憶部（不図示）に記憶する。そして、クエリ実行部13は、クエリの実行の終了を示すクエリ実行終了メッセージをAP2に送信する。

続いて、AP2がフェッチを実行し、その際に、仮想DBサーバ100は、自身に記憶したイベント情報をフェッチに対するデータとして送信する。つまり、AP2は、参照クエリの対する応答として、仮想DBサーバ100からイベント情報（データ）を取得することができる。

#### 【0051】

処理の流れ

50

次に、本発明の第1の実施形態に係る仮想DBシステム1000の処理の流れについて、図5を参照して説明する（適宜図4参照）。

図5は、本発明の第1の実施形態に係る仮想DBシステム1000の処理の流れを示すフローチャートである。

なお、アダプタ30内の仮想DB側IF部310を、図4においては、IF部310と記載する。また、AP2には、仮想DBシステム1000に対するクエリの処理結果を表示するためのGUI（Graphical User Interface）6が接続されているものとする。

#### 【0052】

まず、AP2は、仮想DBシステム1000の仮想DBサーバ100に対して、監視対象装置、監視条件を含む参照クエリを発行する（ステップS10）。

AP2は、例えば、参照クエリとして、「SELECT \* FROM eventTable WHERE host= '192.168.0.10' ;」を発行する。ここで、「host= '192.168.0.10'」は、監視対象装置を識別するためのIPアドレスを表す。そして、この参照クエリは、仮想DBサーバ100に記憶されたテーブルである「eventTable」から、その監視対象装置についての所定の監視条件に合致する情報の取得要求を意味する。監視条件としては、例えば、監視対象装置の状態情報であるCPU使用率やメモリ使用率であること（そのCPU使用率やメモリ使用率が所定の閾値を超えたことを示す情報であること）や、障害発生情報等であること等が設定される。

#### 【0053】

次に、仮想DBサーバ100は、クエリエンジン10により、受信したクエリを解析し、参照先（FROM句で与えられるテーブル）に応じたアダプタ30を決定する。ここでは、仮想DBサーバ100は、FROM句で与えられる、例えば「eventTable」に対応するアダプタ30として、NE5用のアダプタ30Nが決定されたものとする。そして、仮想DBサーバ100は、WHERE句で与えられる監視対象装置と監視条件とを含む監視対象情報を、決定したアダプタ30（30N）に送信する（ステップS11）。

#### 【0054】

アダプタ30（30N）の仮想DB側IF部310は、監視対象情報を受信すると、その監視対象情報を、通知受付部320に出力する（ステップS12）。通知受付部320は、監視対象情報を取得すると、その監視対象情報に含まれる監視対象装置および監視対象条件を登録することにより、イベント通知の監視を開始する。なお、アダプタ30（30N）の仮想DB側IF部310は、通知受付部320に監視対象情報を登録後、通知受付部320がイベント通知（「trap」）を受け付けるまで待機する。

#### 【0055】

次に、NE5により、イベント通知（図5において、「trap」と記載する。）が送信され、アダプタ30（30N）の通知受付部320がそのイベント通知（「trap」）を受け付ける（ステップS13）。通知受付部320は、イベント通知（「trap」）を受け付けると、登録した監視対象情報（監視対象装置および監視条件）に合致するか否かを判定し、合致する場合に、そのイベント通知（「trap」）を仮想DB側IF部310に出力する（ステップS14）。

#### 【0056】

続いて、アダプタ30（30N）の仮想DB側IF部310は、そのイベント通知の情報（イベント情報）を仮想DBサーバ100のクエリ実行部13に送信する（ステップS15）。そして、クエリ実行部13は、取得したイベント通知の情報（イベント情報）を、記憶部（不図示）に記憶しておく。なお、図5においては、このイベント通知の情報（イベント情報）を、参照クエリによりAP2が取得するデータを意味するものとして、「data」と記載している。そして、クエリ実行部13は、クエリ実行の終了を示すクエリ実行終了メッセージをAP2へ送信する（ステップS16）。

#### 【0057】

クエリ実行終了メッセージを受信したAP2は、フェッチによりイベント通知の情報（イベント情報）を、参照クエリに対する応答のデータとして取得する（ステップS17、

10

20

30

40

50

S 1 8 )。そして、A P 2 は、取得したイベント通知の情報 ( イベント情報 ) を G U I 6 に出力する ( ステップ S 1 9 )。

【 0 0 5 8 】

なお、A P 2 からのフェッチ時に、仮想 D B サーバ 1 0 0 において、イベント通知の情報 ( イベント情報 ) が記憶されていない場合には、仮想 D B サーバ 1 0 0 のクエリ実行部 1 3 は、「null」を A P 2 に対し送信する ( ステップ S 2 0、S 2 1 )。

また、A P 2 は、アダプタ 3 0 ( 3 0 N ) による監視を、例えば、exit メッセージを仮想 D B システム 1 0 0 0 に送信することにより終了させることができる ( ステップ S 2 2 ~ S 2 4 )。

【 0 0 5 9 】

このように、本発明の第 1 の実施形態に係る仮想 D B システム 1 0 0 0、および、仮想 D B システムの情報処理方法によれば、データソースから発信されたイベント通知を、クエリ実行時に仮想 D B サーバ 1 0 0 が取得し、A P 2 からの参照クエリに対する応答として、その A P 2 に送信することができる。よって、従来の D B やファイル等の静的なデータソースに加えて、イベント通知を送信するような動的なデータソース ( N E 5 等 ) を含めて仮想 D B システム 1 0 0 0 に統合し情報処理することができる。

【 0 0 6 0 】

< 第 2 の実施形態 >

次に、仮想 D B システム 1 0 0 0 のアダプタ 3 0 が備えるイベント通知処理部 3 0 0 を介して、( パターン 2 ) フェッチ時にイベント情報を取得する方式について具体的に説明する。

【 0 0 6 1 】

図 6 は、本発明の第 2 の実施形態に係る仮想 D B システム 1 0 0 0 A の構成例を示す機能ブロック図である。

図 6 に示した仮想 D B システム 1 0 0 0 A は、図 4 に示した本発明の第 1 の実施形態に係る仮想 D B システム 1 0 0 0 と比べ、仮想 D B サーバ 1 0 0 内のクエリ実行部 1 3 が、クエリ実行部 1 3 A となっていること、アダプタ 3 0 内のイベント通知処理部 3 0 0 がイベント通知処理部 3 0 0 A となっていることが異なる。また、このイベント通知処理部 3 0 0 A 内には、第 1 の実施形態の機能を一部変更した、仮想 D B 側 I F 部 3 1 0 A、通知受付部 3 2 0 A と共に、新たな構成として通知蓄積部 3 1 5 が備えられる。その他の構成については、図 4 に示した仮想 D B システム 1 0 0 0 の構成と同一の機能を備えるものであるため、同一の符号と名称を付し、詳細な説明は省略する。また、図 6 においては、仮想 D B システム 1 0 0 0 A に N E 5 が 1 台のみ接続された例を示しているが、図 1 に示したように、複数の D B 3 やファイルシステム 4、他の N E 5 や A P 2 がデータソースとして接続されていてもよい。

【 0 0 6 2 】

本発明の第 2 の実施形態に係るアダプタ 3 0 は、クエリエンジン 1 0 のクエリ実行部 1 3 A が実行したクエリを、対応するデータソース ( ここでは、N E 5 ) におけるクエリやそのデータソースに適合するプロトコルに基づく操作 ( コマンド ) 等に変換する処理を実行する。そして、このアダプタ 3 0 は、さらに、図 6 に示すイベント通知処理部 3 0 0 A の機能により、データソースから発信されたイベント通知を受け取り、A P 2 からの参照クエリに対する応答として、仮想 D B サーバ 1 0 0 を介して、そのイベント通知を送信する。より詳細には、仮想 D B サーバ 1 0 0 のクエリ実行部 1 3 A は、クエリ実行時に、アダプタ 3 0 のイベント通知処理部 3 0 0 A に対して、監視対象装置 ( ここでは、N E 5 ) と監視条件とを登録し、そのクエリの実行を終了する。イベント通知処理部 3 0 0 A は、登録した監視対象装置から監視条件に合致するイベント通知を受け付けた場合に、そのイベント通知を蓄積しておく。そして、A P 2 からのフェッチ時に、仮想 D B サーバ 1 0 0 がアダプタ 3 0 ( イベント通知処理部 3 0 0 A ) に対して問い合わせを行うことにより、イベント通知の情報 ( イベント情報 ) を取得し、A P 2 に対して送信する。つまり、フェッチ時にイベント通知の情報 ( イベント情報 ) を仮想 D B サーバ 1 0 0 が取得するように

10

20

30

40

50

する。

このイベント通知処理部 300A は、図 6 に示すように、仮想 DB 側 I F 部 310A と、通知蓄積部 315 と、通知受付部 320A とを備える。

【0063】

仮想 DB 側 I F 部 310A は、仮想 DB サーバ 100 のクエリ実行部 13A から、クエリ形式で取得した監視対象装置（ここでは、NE5）および監視条件の情報（「監視対象情報」）を解釈し、通知受付部 320A に登録することにより、監視を開始させる。

また、仮想 DB 側 I F 部 310A は、仮想 DB サーバ 100（クエリ実行部 13A）からの要求に応じて、通知蓄積部 315 に問い合わせを行い、蓄積されたイベント通知を取得して、仮想 DB サーバ 100（クエリ実行部 13A）にイベント通知の情報（イベント情報）として送信する。

10

【0064】

通知蓄積部 315 は、通知受付部 320A からイベント通知を取得して蓄積（記憶）する。また、通知蓄積部 315 は、蓄積したイベント通知を仮想 DB 側 I F 部 310A からの問い合わせに応じて出力する。

【0065】

通知受付部 320A は、仮想 DB 側 I F 部 310A からの要求により、監視対象装置および監視条件を登録する。そして、通知受付部 320A は、登録した監視対象装置からイベント通知を受け付けた場合に、監視条件に合致する否かを判定し、監視条件に合致する場合に、そのイベント通知を通知蓄積部 315 に出力する。

20

【0066】

仮想 DB サーバ 100（クエリ実行部 13A）は、アダプタ 30 の通知受付部 320A による監視対象情報（監視対象装置および監視条件）の登録が完了した場合に、クエリ実行の終了を示すクエリ実行終了メッセージを AP2 へ送信する。また、クエリ実行部 13A は、AP2 からのフェッチによるデータ読み出し要求があったときに、アダプタ 30（仮想 DB 側 I F 部 310A）に対し、イベント通知の情報（イベント情報）の問い合わせを行い、通知蓄積部 315 に新規のイベント通知の情報（イベント情報）が蓄積されていれば取得し、そのフェッチの応答として、AP2 へ送信する。

【0067】

処理の流れ

30

次に、本発明の第 2 の実施形態に係る仮想 DB システム 1000A の処理の流れについて、図 7 を参照して説明する（適宜図 6 参照）。

図 7 は、本発明の第 2 の実施形態に係る仮想 DB システム 1000A の処理の流れを示すフローチャートである。

なお、アダプタ 30 内の仮想 DB 側 I F 部 310A を、図 4 においては、I F 部 310A と記載している。また、AP2 には、仮想 DB システム 1000A に対するクエリの処理結果を表示するための GUI6 が接続されているものとする。

【0068】

まず、AP2 は、仮想 DB システム 1000A の仮想 DB サーバ 100 に対して、監視対象装置、監視条件を含む参照クエリを発行する（ステップ S30）。

40

AP2 は、例えば、参照クエリとして、「SELECT \* FROM eventTable WHERE host= '192.168.0.10';」を発行する。ここで、「host= '192.168.0.10'」は、監視対象装置を識別するための IP アドレスを表す。そして、この参照クエリは、仮想 DB サーバ 100 に記憶されたテーブルである「eventTable」から、その監視対象装置についての所定の監視条件に合致する情報の取得要求を意味する。監視条件としては、例えば、監視対象装置の状態情報である CPU 使用率やメモリ使用率であること（その CPU 使用率やメモリ使用率が所定の閾値を超えたことを示す情報であること）や、障害発生情報等であること等が設定される。

【0069】

次に、仮想 DB サーバ 100 は、クエリエンジン 10 により、受信したクエリを解析し

50

、参照先（FROM句で与えられるテーブル）に応じたアダプタ30を決定する。ここでは、仮想DBサーバ100は、FROM句で与えられる「eventTable」に対応するアダプタ30として、NE5用のアダプタ30Nが決定されたものとする。そして、仮想DBサーバ100は、WHERE句で与えられる監視対象装置と監視条件と含む監視対象情報を、決定したアダプタ30（30N）に送信する（ステップS31）。

#### 【0070】

アダプタ30（30N）の仮想DB側IF部310Aは、監視対象情報を受信すると、その監視対象情報を、通知受付部320Aに出力する（ステップS32）。通知受付部320Aは、監視対象情報を取得すると、その監視対象情報に含まれる監視対象装置および監視対象条件を登録することにより、イベント通知の監視を開始する。

10

#### 【0071】

続いて、通知受付部320Aは、監視対象情報の登録完了メッセージを、仮想DB側IF部310Aに送信し（ステップS33）、仮想DB側IF部310Aが、仮想DBサーバ100のクエリ実行部13Aに、その登録完了メッセージを転送する（ステップS34）。そして、アダプタ30から登録完了メッセージを受信したクエリ実行部13Aは、クエリ実行の終了を示すクエリ実行終了メッセージをAP2へ送信する（ステップS35）。

#### 【0072】

次に、NE5により、イベント通知（図7において、「trap」と記載する。）が送信され、アダプタ30（30N）の通知受付部320Aがそのイベント通知（「trap」）を受け付ける（ステップS36）。通知受付部320Aは、イベント通知（「trap」）を受け付けると、登録した監視対象情報（監視対象装置および監視条件）に合致するか否かを判定し、合致する場合に、そのイベント通知（「trap」）を通知蓄積部315に出力し（ステップS37）、通知蓄積部315においてそのイベント通知（「trap」）が蓄積（記憶）される。

20

#### 【0073】

一方、クエリ実行終了メッセージを受信したAP2は、フェッチによるイベント通知の情報（イベント情報）の読み出し要求を送信する（ステップS38）。仮想DBサーバ100のクエリ実行部13Aは、フェッチによる読み出し要求を受信すると、アダプタ30の仮想DB側IF部310Aに問い合わせを行い（ステップS39）、仮想DB側IF部310Aは、その問い合わせを通知蓄積部315に転送する（ステップS40）。

30

#### 【0074】

通知蓄積部315は、その問い合わせを受信した際に、対応するイベント通知（「trap」）を記憶している場合には、そのイベント通知（「trap」）を仮想DB側IF部310Aに送信する（ステップS41）。そして、仮想DB側IF部310Aは、そのイベント通知の情報（イベント情報）を仮想DBサーバ100のクエリ実行部13Aに送信する（ステップS42）。クエリ実行部13Aは、取得したイベント通知の情報（イベント情報）をフェッチによる読み出し処理としてAP2へ送信する（ステップS43）。このイベント通知の情報（イベント情報）を受信したAP2は、取得したイベント情報をGUI6に出力する（ステップS44）。

40

#### 【0075】

そして、AP2は、繰り返しフェッチを実行することにより、ステップS38～S44と同様の処理を行う。具体的には、アダプタ30は、ステップS45およびS46の処理によりイベント通知の情報（イベント情報）を通知蓄積部315に蓄積（記憶）し、AP2のフェッチ時に、仮想DBサーバ100のクエリ実行部13Aが、アダプタ30（仮想DB側IF部310A、通知蓄積部315）に問い合わせることにより、イベント情報を取得し、GUI6に出力することができる（ステップS47～S53）。

#### 【0076】

一方、符号（A-1）に示すように、ステップS54～S56の処理により、アダプタ30の通知蓄積部315に対し、フェッチ時に伴う問い合わせを行った場合に、通知蓄積

50

部 3 1 5 がイベント通知 ( 「 trap 」 ) を記憶していなかったときには、通知蓄積部 3 1 5 は、仮想 DB 側 I F 部 3 1 0 A に対し 「 null 」 を送信する ( ステップ S 5 7 ) 。 「 null 」 を受信した仮想 DB 側 I F 部 3 1 0 A は、その 「 null 」 を仮想 DB サーバ 1 0 0 のクエリ実行部 1 3 A に送信し ( ステップ S 5 8 ) 、クエリ実行部 1 3 A が、フェッチに対する応答として 「 null 」 を A P 2 へ送信する ( ステップ S 5 9 ) 。

【 0 0 7 7 】

また、A P 2 は、アダプタ 3 0 ( 3 0 N ) による監視を、例えば、exit メッセージを仮想 DB システム 1 0 0 0 A に送信することにより終了させることができる ( ステップ S 6 0 ~ S 6 2 ) 。

【 0 0 7 8 】

このように、本発明の第 2 の実施形態に係る仮想 DB システム 1 0 0 0 A 、および、仮想 DB システムの情報処理方法によれば、データソースから発信されたイベント通知を、フェッチ時に仮想 DB サーバ 1 0 0 がアダプタ 3 0 から取得し、A P 2 からの参照クエリに対する応答として、その A P 2 に送信することができる。よって、従来の DB やファイル等の静的なデータソースに加えて、イベント通知を送信するような動的なデータソース ( N E 5 等 ) を含めて仮想 DB システム 1 0 0 0 に統合し情報処理することができる。

【 0 0 7 9 】

< 第 2 の実施形態の変形例 >

第 2 の実施形態に係る仮想 DB システム 1 0 0 0 A においては、図 7 の符号 ( A - 1 ) に示すように、アダプタ 3 0 の通知蓄積部 3 1 5 が、イベント通知 ( 「 trap 」 ) を記憶していなかった場合には、「 null 」 を仮想 DB サーバ 1 0 0 および A P 2 側に送信していた。これに対し、第 2 の実施形態の変形例として、通知蓄積部 3 1 5 が、仮想 DB サーバ 1 0 0 からの問い合わせを受信したときに、イベント通知 ( 「 trap 」 ) を記憶していなかった場合、N E 5 からのイベント通知 ( 「 trap 」 ) を通知受付部 3 2 0 A が受け付け、通知蓄積部 3 1 5 に記憶されるまで待機するようにしてもよい。この場合、図 8 の符号 ( A - 2 ) に示すように、ステップ S 7 0 ~ S 7 2 の処理により、問い合わせを受信したアダプタ 3 0 の通知蓄積部 3 1 5 は、「 null 」 を送信するのではなく、次のイベント通知 ( 「 trap 」 ) を通知受付部 3 2 0 A が受け付けて、通知蓄積部 3 1 5 自身が記憶するまで待機する。そして、イベント通知 ( 「 trap 」 ) を受信し記憶したことを契機として ( ステップ S 7 3 、 S 7 4 ) 、通知蓄積部 3 1 5 が、そのイベント通知 ( 「 trap 」 ) を仮想 DB 側 I F 部 3 1 0 A に送信し、仮想 DB サーバ 1 0 0 のクエリ実行部 1 3 A を介して、A P 2 へ送信するようにしてもよい ( ステップ S 7 5 ~ S 7 7 ) 。

【 0 0 8 0 】

このようにすることにより、仮想 DB システム 1 0 0 0 A は、アダプタ 3 0 の通知蓄積部 3 1 5 にイベント通知 ( 「 trap 」 ) が記憶されていない場合の無駄となる処理 ( 図 7 のステップ S 5 4 ~ S 5 9 の処理 ) をなくすることができる。よって、仮想 DB サーバ 1 0 0 および A P 2 の処理負荷を低減することができる。

【 0 0 8 1 】

また、第 2 の実施形態のその他の変形例として、例えば、以下のようにすることもできる。第 2 の実施形態に係る仮想 DB システム 1 0 0 0 A においては、図 7 に示すように、A P 2 からのフェッチを仮想 DB サーバ 1 0 0 が受信する度に、仮想 DB サーバ 1 0 0 がアダプタ 3 0 の通知蓄積部 3 1 5 に対して問い合わせを行うようにしていた ( 例えば、ステップ S 3 8 ~ S 4 0 ) 。これに対し、他の変形例として、仮想 DB サーバ 1 0 0 がフェッチの受信の度に問い合わせを行うのではなく、仮想 DB サーバ 1 0 0 のクエリ実行部 1 3 A が、所定の時間間隔で、通知蓄積部 3 1 5 に問い合わせを行い、その問い合わせの際に、イベント通知の情報 ( イベント情報 ) を取得し仮想 DB サーバ 1 0 0 内に記憶しておくようにしてもよい。このようにすることにより、仮想 DB サーバ 1 0 0 は、A P 2 からのフェッチの受信時に、アダプタ 3 0 に問い合わせることなく、自身に記憶されたイベント情報を、A P 2 に送信することができる。

【 0 0 8 2 】

10

20

30

40

50

以上説明したように、本実施形態に係る仮想DBシステム1000(1000A)、および、仮想DBシステムの情報処理方法によれば、従来のDBやファイル等の静的なデータソースに加えて、イベント通知を送信するような動的なデータソース(NE5等)を含めて仮想DBに統合し情報処理することができる。

また、従来の仮想DBと同様に、AP2からの参照クエリに対する応答として、動的なデータソースからのイベント通知をAP2に送信できるようにしたことにより、AP2の変更を極力なくすことができ、データソースからのイベント通知を従来のDBアクセスと同様に扱うことができる。さらに、仮想DBサーバ100の構成を変更せず、アダプタ30にイベント通知処理部300を備える構成としたことにより、従来の仮想DBサーバ100の改編を行うことなく、データソースからのイベント通知を扱うことができる。

10

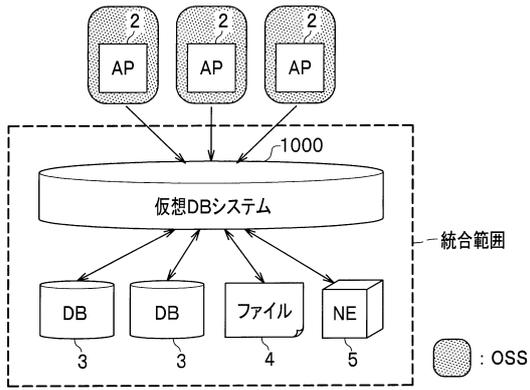
【符号の説明】

【0083】

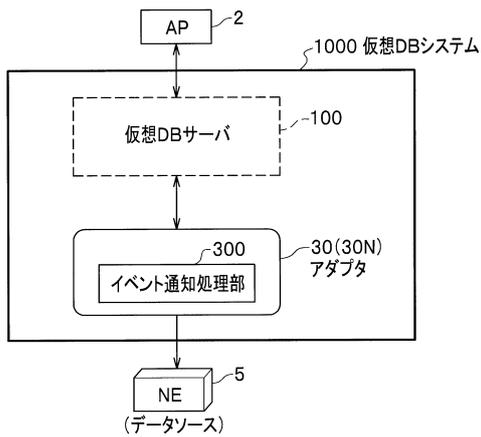
- 1, 1000, 1000A 仮想DBシステム
- 2 AP(クライアントAP)
- 3 DB
- 4 ファイル(ファイルシステム)
- 5 NE
- 6 GUI
- 10 クエリエンジン
- 11 クエリ解析・書換部
- 12 クエリ最適化部
- 13, 13A クエリ実行部
- 20 スキーマ・ルール格納部
- 30 アダプタ
- 100 仮想DBサーバ
- 300, 300A イベント通知処理部
- 310, 310A 仮想DB側IF部
- 315 通知蓄積部
- 320, 320A 通知受付部

20

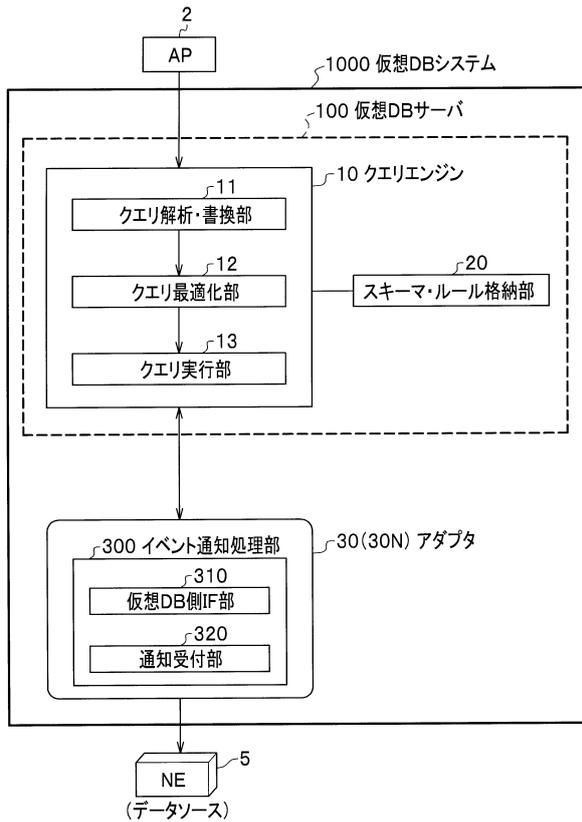
【図1】



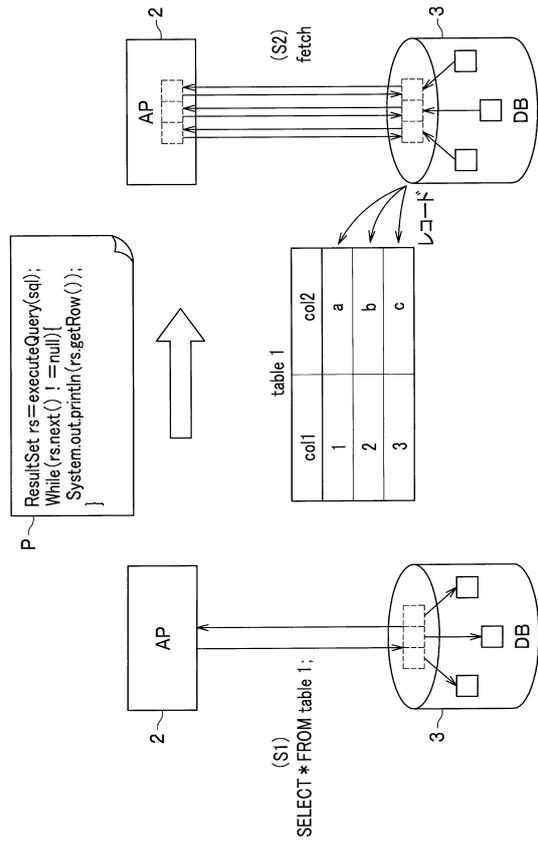
【図2】



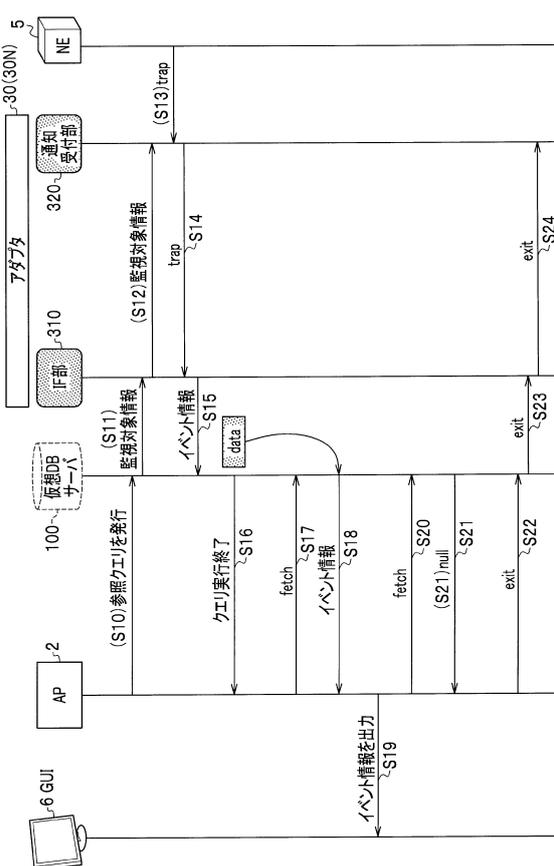
【図4】



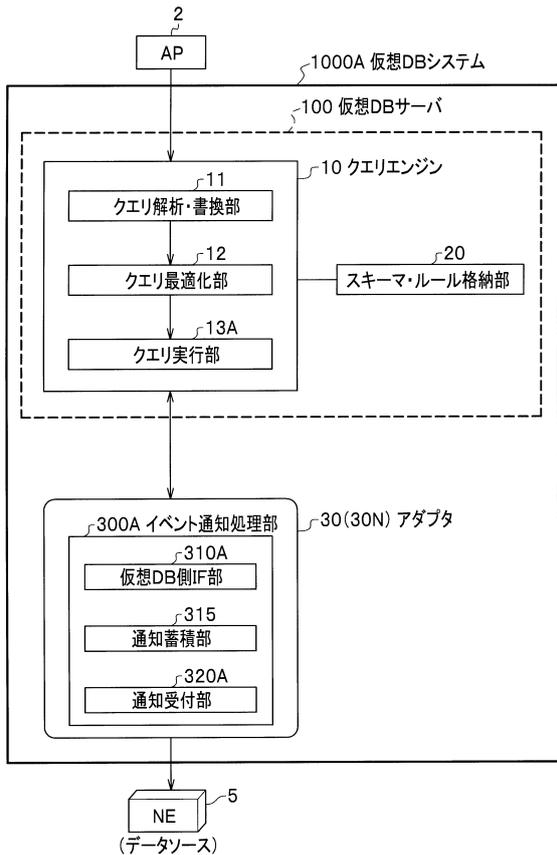
【図3】



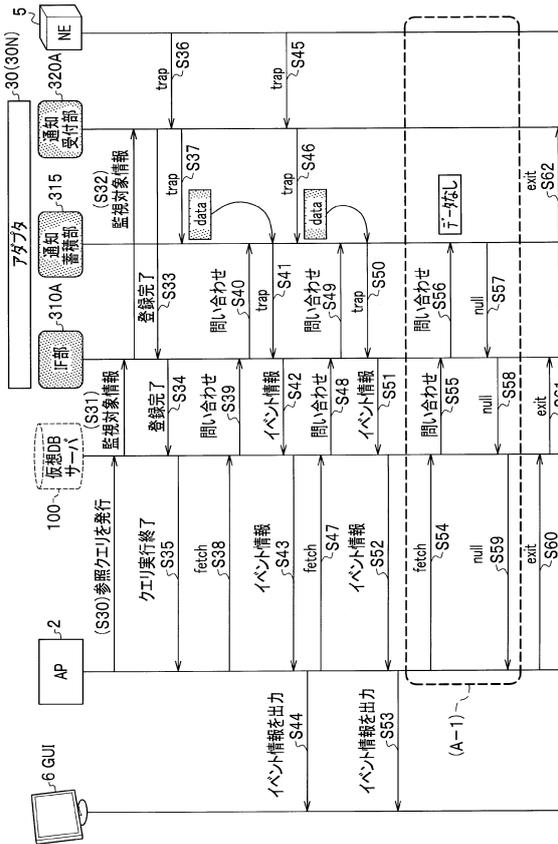
【図5】



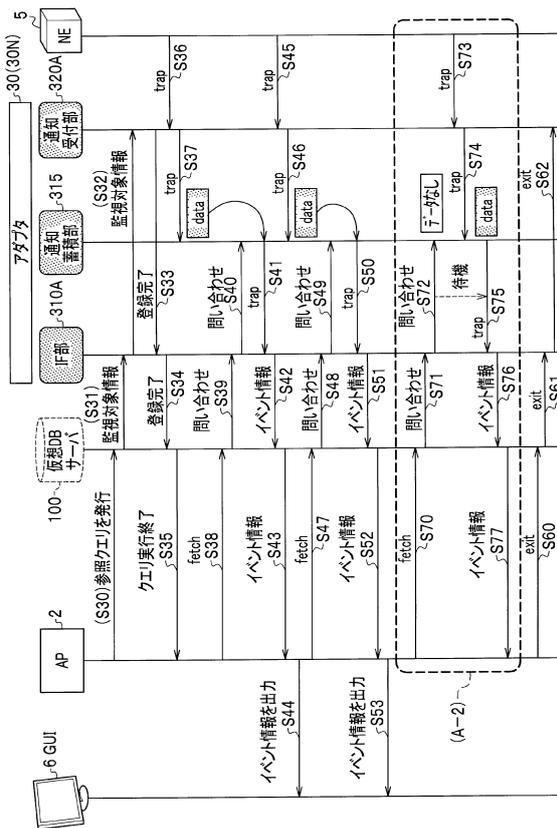
【図6】



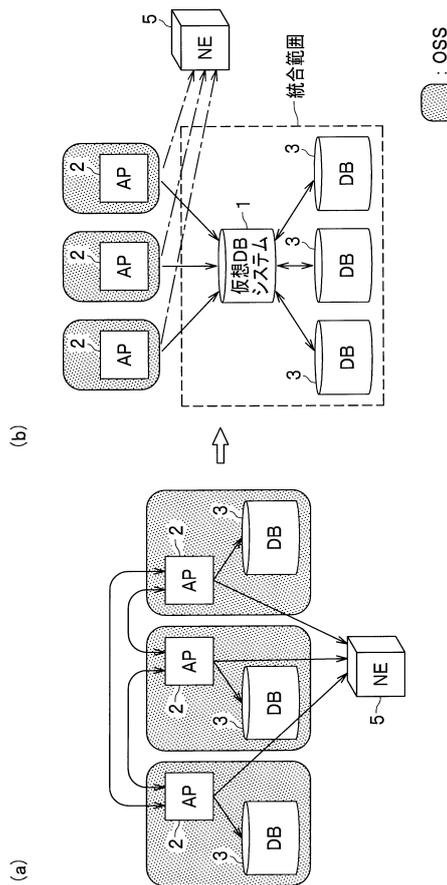
【図7】



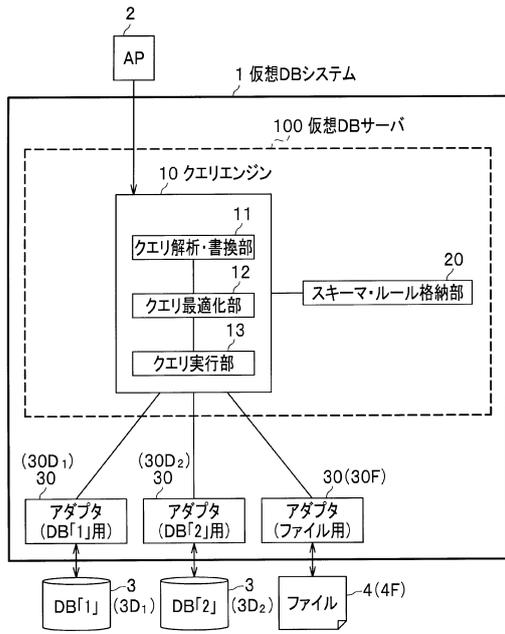
【図8】



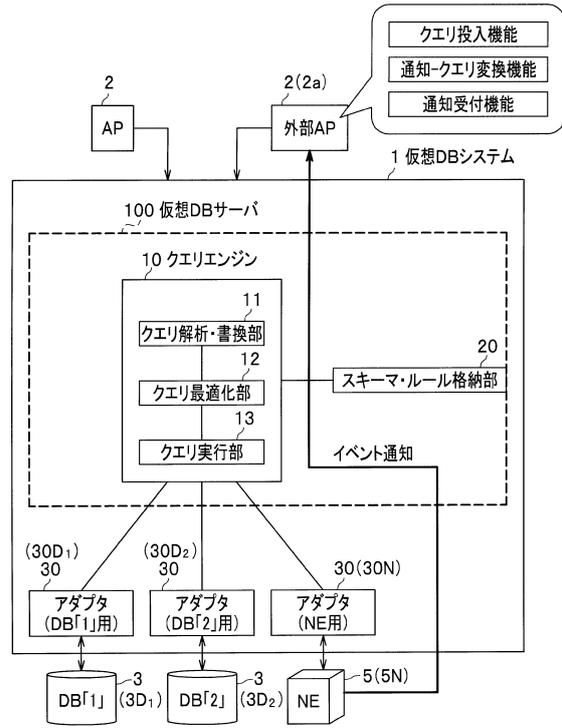
【図9】



【図10】



【図11】



---

フロントページの続き

- (56)参考文献 特開2006-195788(JP,A)  
特開2006-004017(JP,A)  
特開2001-101065(JP,A)  
米国特許出願公開第2008/0275956(US,A1)  
武 直樹 他2名, “仮想DBを用いたOSSデータ統合の実現性”, 電子情報通信学会技術研究報告 IEICE Technical Report, 日本, 一般社団法人 電子情報通信学会, 2013年 3月 7日, 第112巻, 第492号, PP. 71-76, 【ISSN】0913-5685

(58)調査した分野(Int.Cl., DB名)

G06F 12/00  
G06F 17/30  
G06F 13/00  
H04L 12/24 - 12/26