

## 第4章 京都府

---

京都府において特徴的な実証実験について詳細に説明する。

## 4.1 データセンター機能実証

### 4.1.1 利用拠点バックアップ（市町村バックアップ）

#### （1）実証の概要・目的

信頼性の高い広域ネットワークといえども、障害や、運用・保守による回線切断が発生しうる。可能性は低いもののデータセンターの火災等によるデータセンターの一時的な利用停止も想定される。自治体クラウドは、データセンターに集約したアプリケーションを利用して業務を行うものである。今までは、庁舎内に構築したシステムを利用していたためこれらの事象について考慮する必要がなかったが、自治体クラウドにおいてこうした事態を想定した取組が必要である。

そこで、本実証では、業務中に回線障害等によりデータセンターのアプリケーションが利用できなくなることを想定し、利用拠点側に用意した環境において運用を継続する利用拠点バックアップの実証を行った。

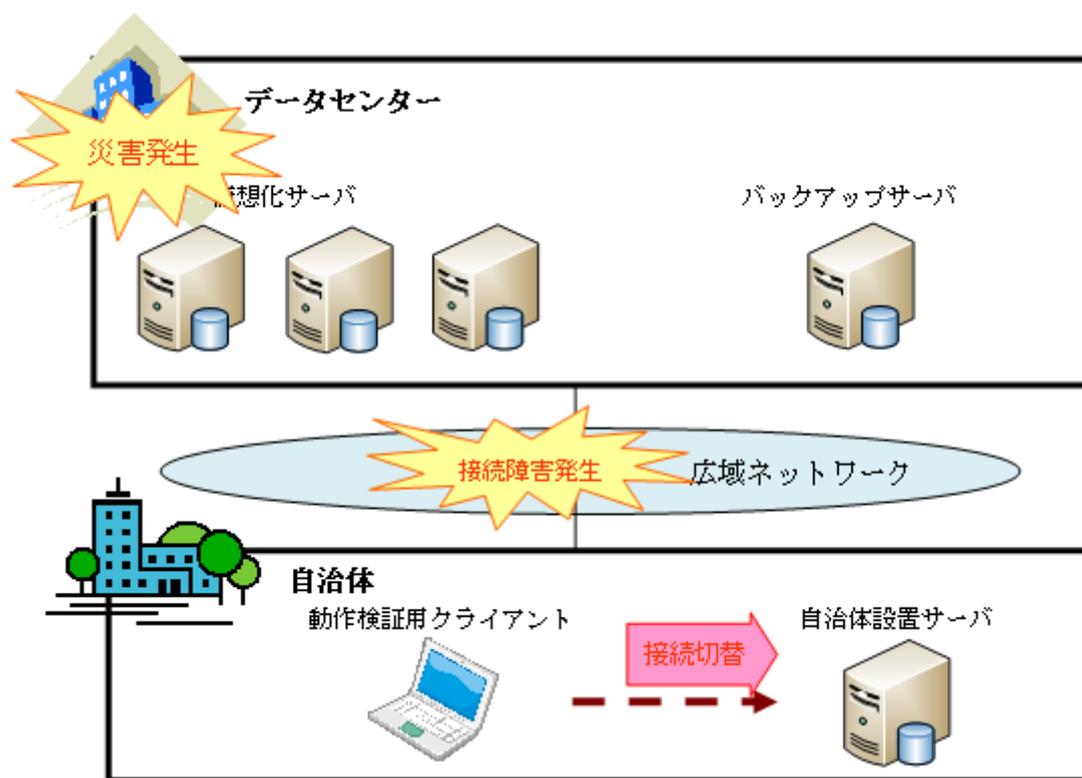


図 4-1 利用拠点バックアップ（市町村バックアップ）イメージ

## (2) 実証の内容

### ア) 前提条件

当実証実験において、本番システムのデータベースやモジュール等については、全て1自治体を単位として管理する事とし（例えば1データベース中には必ず1自治体分の環境のみが存在する）、確認するのは、1自治体分のみ環境を対象とする。なお、縮退運転環境（※）も同様である。

#### ※縮退運転環境とは

本番システムのバックアップをデータセンターだけではなく自治体側に設定したサーバにも置くことによりシステムの冗長化を図るものである。バックアップデータにより、本番環境と同じものが自治体側に置かれたサーバに構築される意味では「本番環境のコピー」というイメージになる。毎日のバックアップはタスクスケジュールにより深夜に実行され、その後システムのアプリケーションや各種設定を自動で行う処理がタスクスケジュールにより実行され、翌朝には前日の本番システムと同内容の環境が自治体側に出来あがる仕組みである。その環境はすぐに起動できる状態で構築が完了しているが、待機している状態になっている。なお本番システムをコピーした環境とはいえ、その内容は前日のものであるため、縮退運転環境を使用するタイミングによっては実際の情報とは差異が生まれるところが運用に影響する。

また、縮退運転環境は、タスクスケジュールに登録されたバッチファイルにより、下記の順序で毎日、深夜に構築される。

- ① 本番システムのデータベースをエクスポートし、そのダンプファイルをバックアップとして自治体側へコピーする。
- ② 本番システムで実装されている「オンラインモジュール」、「バッチモジュール」、「帳票定義」「クライアントモジュール」を自治体側へコピーする。
- ③ 上記①②の作業により自治体側へ保管されたバックアップから、「縮退運転環境」を構築する。
- ④ クライアントマシンについては、本番システム、縮退運転環境の二つの環境をすぐに起動できるような状態になっており、上記③の状態では、このどちらも起動できる状態にある。しかし、本番システムが正常に稼働している時に、縮退運転環境を起動する必要がない（また、誤操作の可能性があるので）ため、縮退運転環境を、一時的に起動できない状態に変更しておく。

### イ) 実施環境

実証実験を実施した環境は以下のとおりである。

A. 環境

当実証実験において、データセンター側の本番システム環境は下記3台のサーバ（仮想マシン）を使用する。

- アプリケーションサーバ・・・「オンラインモジュール」を実装
- バッチ帳票サーバ・・・「バッチモジュール」、「帳票定義」、「クライアントモジュール」を実装
- データベースサーバ・・・「データベース」を実装

一方、縮退運転環境については、上記3台のサーバに実装されている内容を1台のサーバに集約する環境とする。

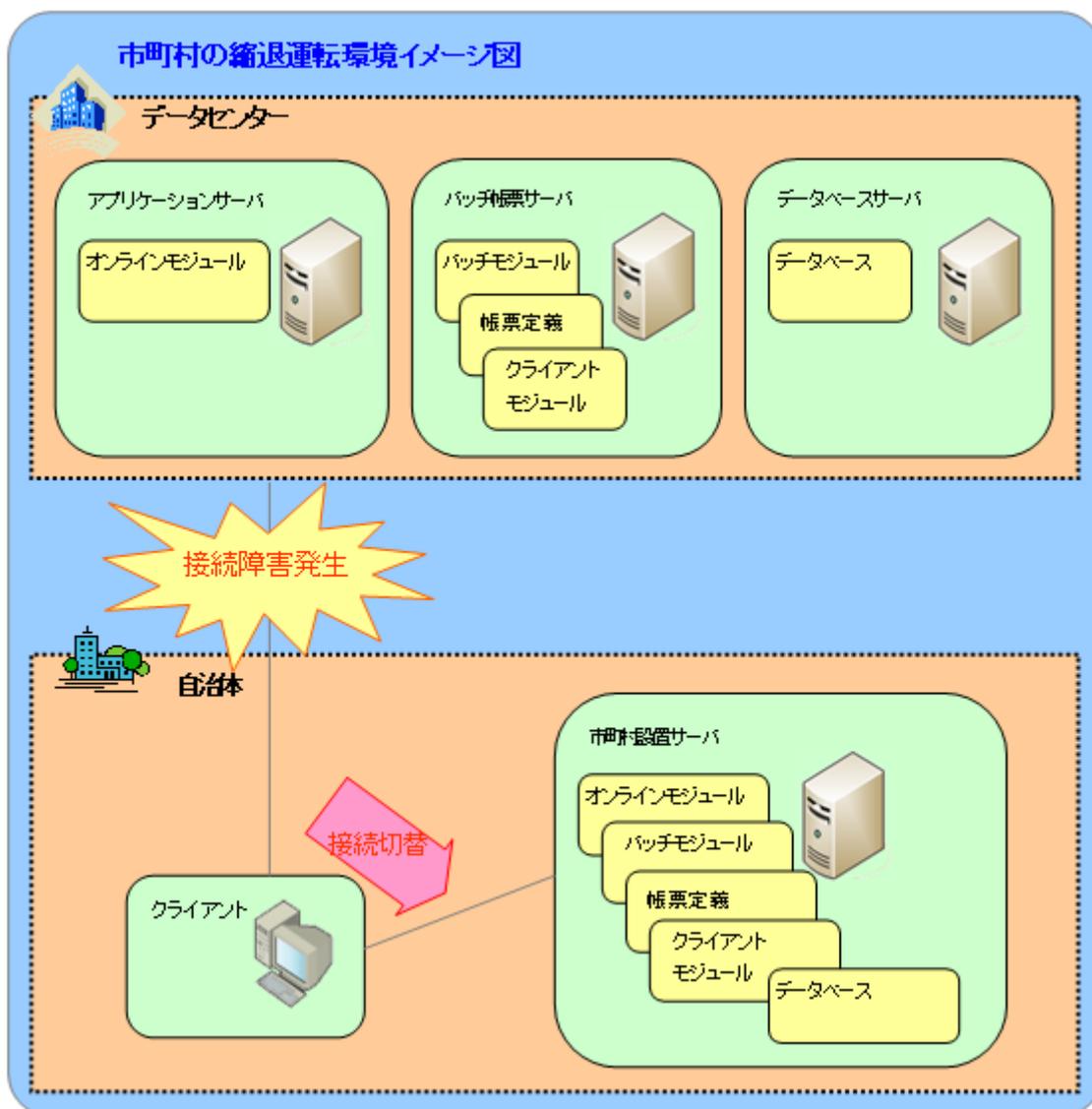


図 4-2 市町村の縮退運転環境イメージ

## B. 使用システム・使用業務

基幹業務支援システム（共通）の全業務。

## C. 使用データ

約55,000人規模の自治体の住民情報システムデータを使用した。

## D. 使用ハードウェア

実証環境で使用したサーバ及びクライアントは以下のとおりである。

表 4-1 市町村バックアップ使用ハードウェア一覧

No	名称	台数	スペック			
			CPU	メモリ	HDD	OS
1	バッチ・帳票サーバ （仮想マシンとして用意）	1	Intel®Xeon® E5503 2GHz	4GB	C: 30GB D: 30GB	Windows Server 2008 Standard(64bit)
2	アプリケーションサーバ （仮想マシンとして用意）	3	Intel®Xeon® E5503 2GHz	4GB	C: 30GB D: 30GB	Windows Server 2008 Standard(64bit)
3	データベースサーバ （仮想マシンとして用意）	1	Intel®Xeon® E5503 2GHz	4GB	C: 30GB D: 30GB E: 250GB	Windows Server 2008 Standard(64bit)
4	クライアント	1	Intel®Core i5	4GB	C: 160GB B	Windows7 Professional(32bit)
5	縮退運転サーバ （市町村設置サーバ）	1	IntelPentium プロセッサ - G6950 2.80GHz	13GB	C: 50GB D: 30GB E: 327GB	Windows Server 2008 Standard(64bit)

## E. 使用ソフトウェア

実証環境で使用したソフトウェアは以下のとおりである。

表 4-2 オンサイトバックアップ使用ソフトウェア一覧

No	名称	使用用途
1	DF	ファイル比較用ツール

## ウ) 実施手順

前述のように、クライアントがある自治体から遠く離れたデータセンター側に本番システムが実装されている自治体クライアントの環境において、データセンターと自治体間の通信障害は、本番システムが完全停止となる深刻な事態となる。このようなデータセンター側の本番システムが使用できない状況に備えて、自治体側にも「常に本番システムと同じ環境」を置き、障害が発生した際に代行運用を行うのが縮退運転環境の主目的である。

このため、縮退運転環境は、毎日、本番システムの環境と同じになるように更新され続けなければならない。当実証実験では、タスクスケジュールによって実行されるバッチ処理によって、データセンター側の本番システムと同じ環境が構築されるか、またシステムを問題なく使用できるかについて確認した。

### A. 本番システムのデータベースが縮退運転環境に反映されているかを確認する。

タスクスケジュールに登録したバッチファイルの実行により、「本番システムのデータベースをバックアップしたダンプファイル」をデータセンター側から自治体設置サーバ側へコピーし、その後、縮退運転環境へインポートを行って縮退運転環境のデータベースを作成する。

当実証実験では、「データベースに存在するテーブル内のデータ件数」について、本番システムと縮退運転環境とを比較し、内容が一致しているかを確認する。

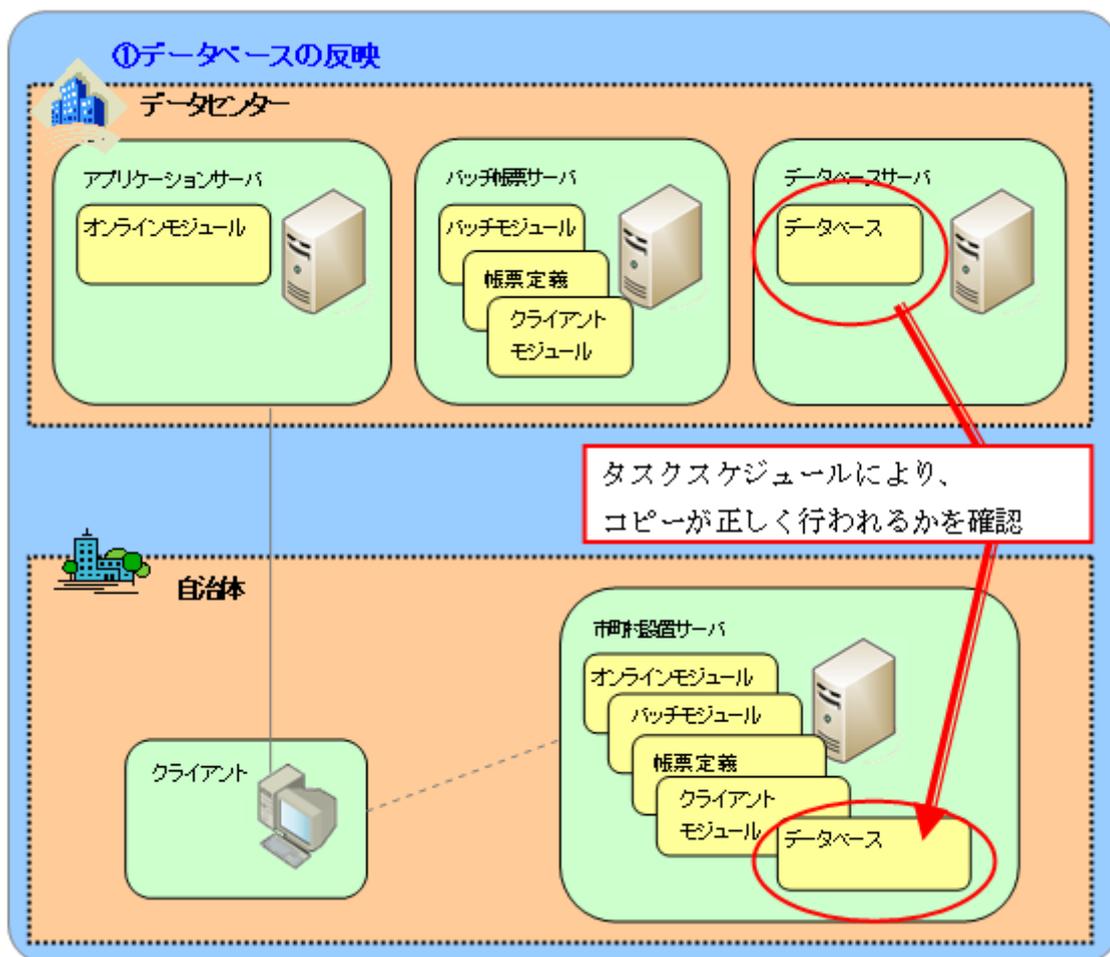


図 4-3 市町村バックアップのデータベースの反映

B. 本番システムのモジュールが、縮退運転環境に反映されているかを確認する。

タスクスケジュールに登録したバッチファイルの実行により、本番システムの「オンラインモジュール」、「バッチモジュール」「帳票定義」「クライアントモジュール」をセンター側から自治体設置サーバ側へコピーし、その後、縮退運転環境のシステムとして配備する。

当実証実験では、本番システム、縮退運転環境に配備されているモジュールを比較し、内容が一致しているかを確認する。

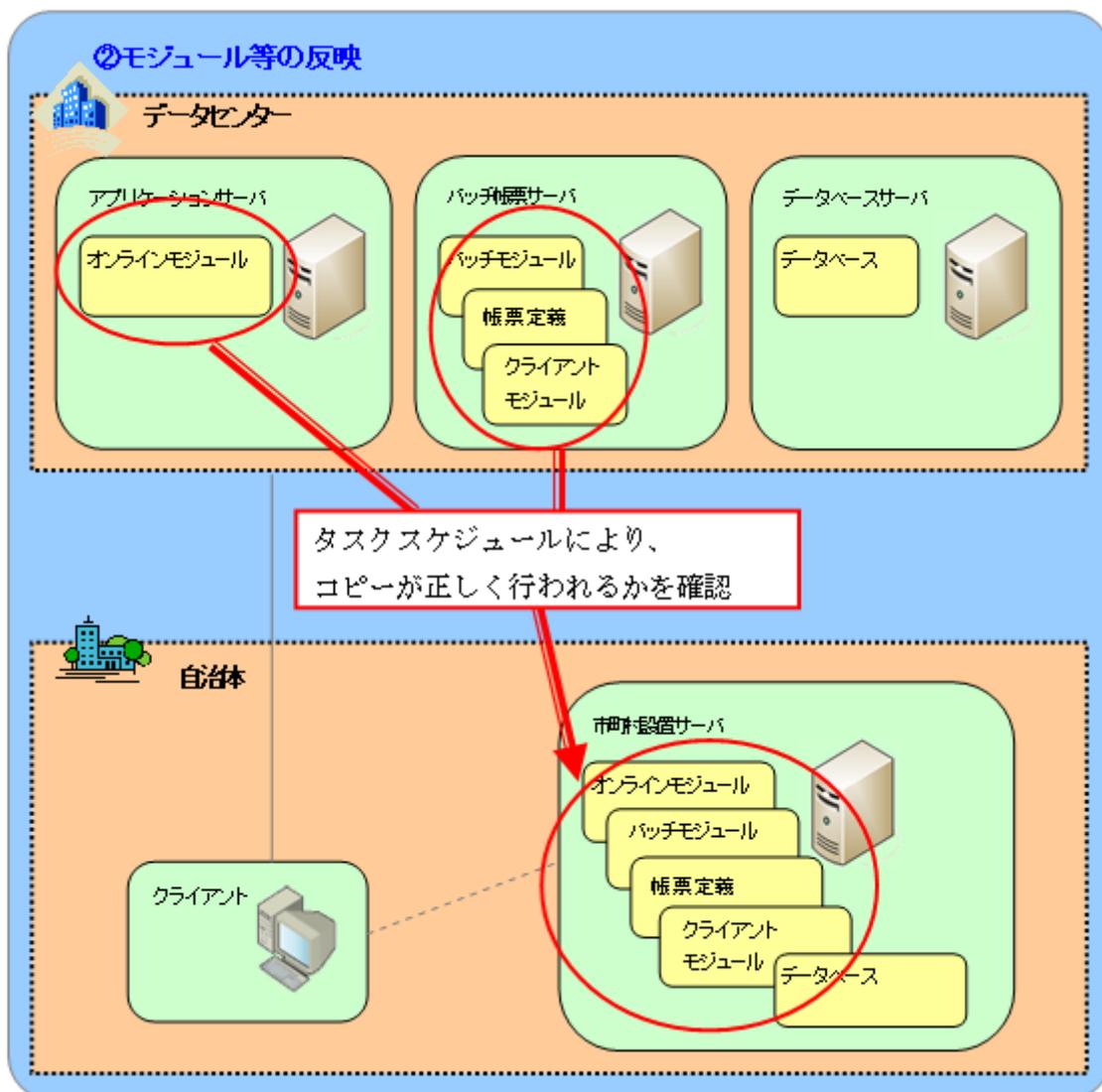


図 4-4 市町村バックアップのモジュール等の反映

C. 通常、縮退運転環境ではシステムが起動しないことを確認する。

データセンター側で本番システムが稼動している通常時に、縮退運転環境が起動可能な状態にあると誤操作の原因（※1）になる。このため、システムが起動しないように、縮退運転環境のシステムを構築後、システム起動に必要なテーブルについてテーブル名を変更する。

当実証実験では、バッチ処理により構築された直後の縮退運転環境でシステムを起動し、システムが正常に起動しない（※2）ことを確認する。

※1 例えば、本番システムを操作して台帳を登録しているつもりが、縮退運転環境を操作してしまい、実際の本番システムでは、台帳の登録漏れが発生する等。

※2 起動時に必要なテーブルが見つからないため、エラーが発生す

ることになる。

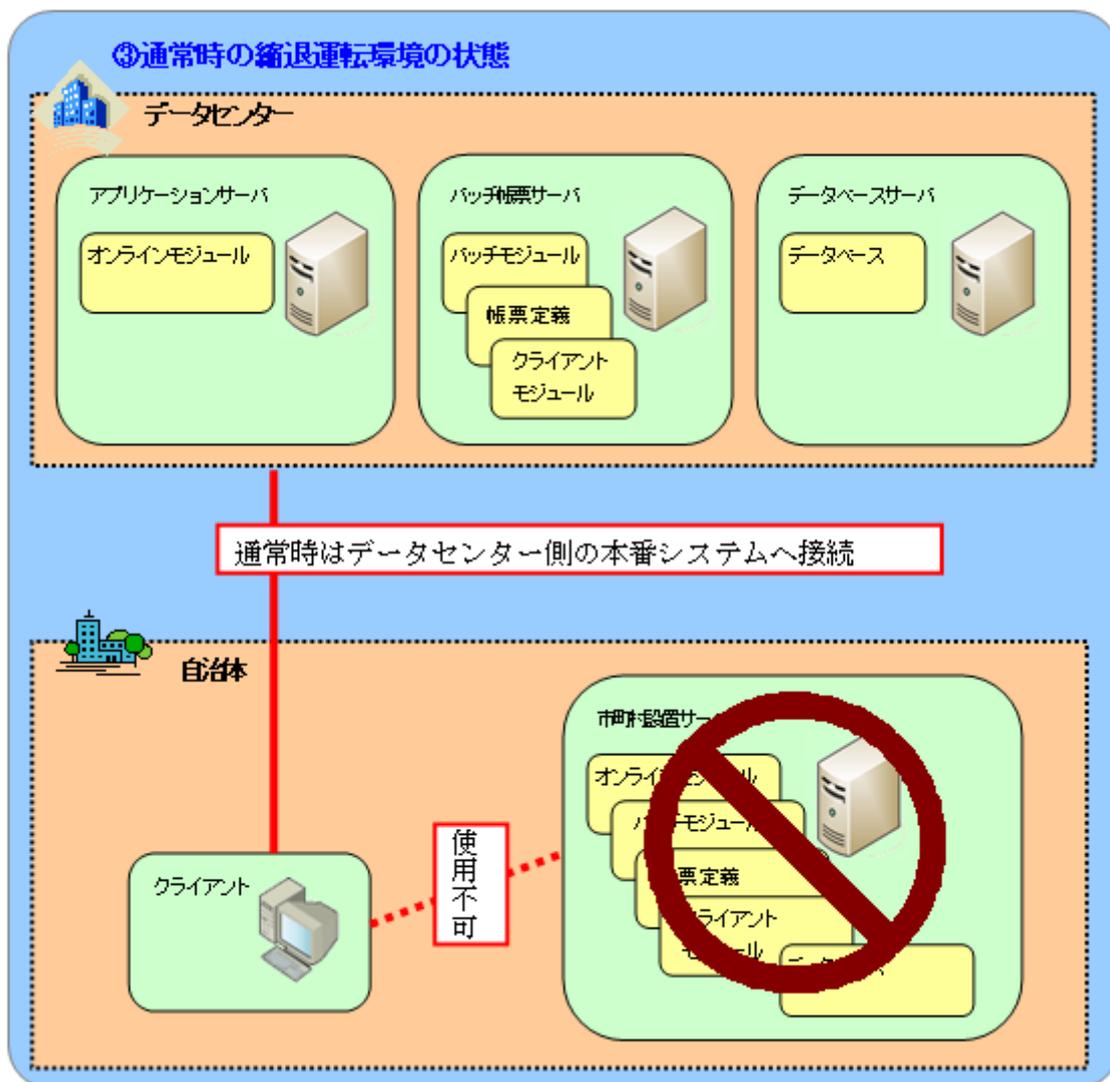


図 4-5 通常時の縮退運転環境の状態

- D. 接続障害が発生したと考え、手動動作により縮退運転環境を起動し、システムが正常に動作することを確認する。

バッチファイルを実行することにより、縮退運転環境は起動可能な状態に切り替わる。その後、システムを起動すると、本番システムと全く同じ機能を提供できる状態になる。当実証実験では、縮退運転環境を起動可能な状態に切り替えた後、システムが正常に稼動するかを下記項目の内容で確認する。

起動時に必要なテーブルを元の状態に戻す（テーブル名を基に戻す）。

表 4-3 縮退運転での動作確認テスト項目

No	テスト項目
1	トップページにアクセスできること
2	クライアントモジュールのバージョンが最新であること
3	メンテナンス画面にログインできること
4	システムにログインし、総合メニューが表示されること
5	想定したデータベースサーバに接続されていること
6	想定したアプリケーションサーバに接続されていること
7	即時帳票が正しくプレビューできること
8	バッチ処理(スプールあり)が正しく実行できること
9	スプール帳票が閲覧できること
10	端末使用状況が確認できること
11	ヘルプファイルが表示できること

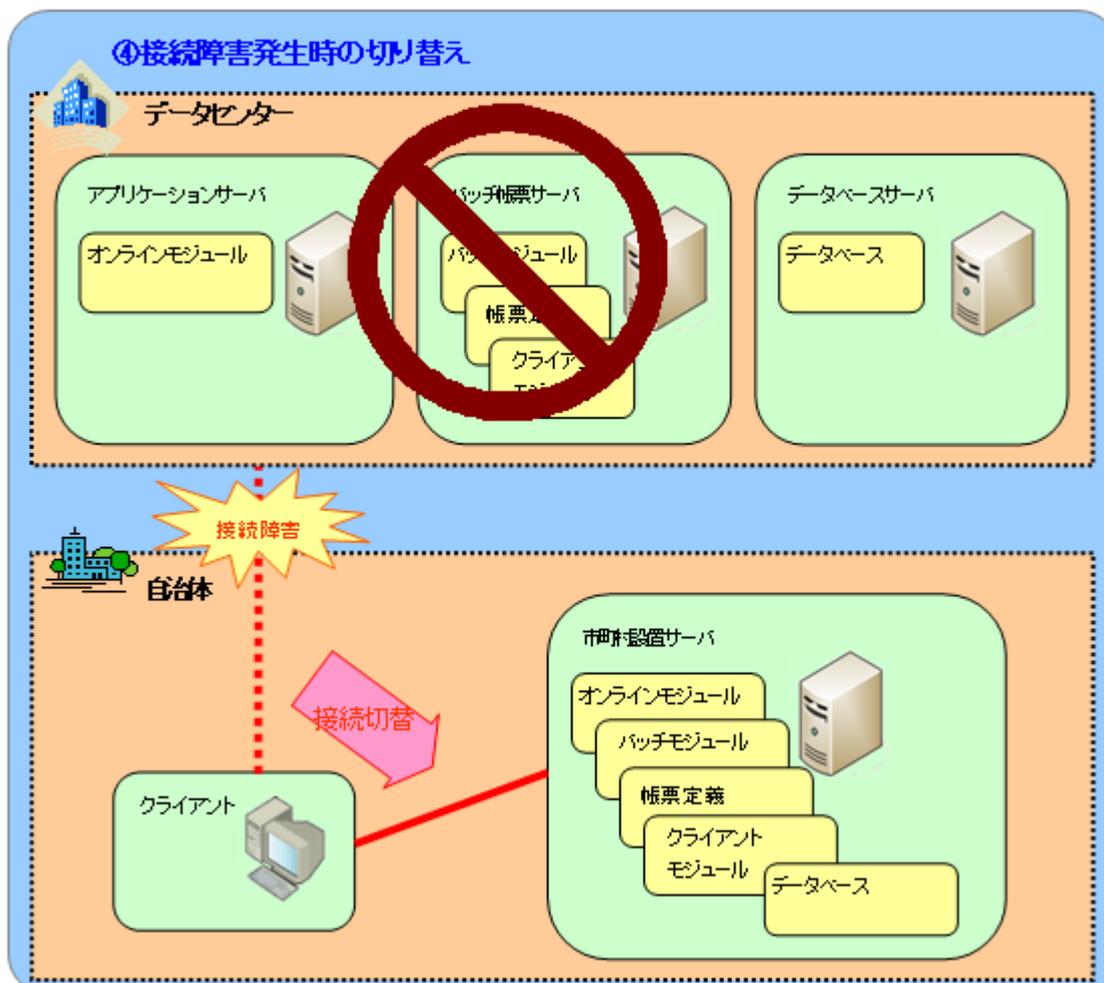


図 4-6 接続場外発生時の切替

- E. 障害が解消したと想定し、縮退運転環境を起動不可の状態にし、センター運用ができることを確認する。

障害が発生していたセンター側の本番システムが復旧した場合、速やかに縮退運転環境システムを起動不可能な状態に切り替える必要がある。

ここでは、バッチファイルの実行により起動に必要なテーブルの名前を変更し、縮退運転環境のシステムを起動不可の状態に戻す。その後データセンター接続に切り替える。

また、縮退運転環境下で更新された証明書発行ログや発布番号を目視確認し、差分についてはデータセンター側のデータへ手動にて反映させる。

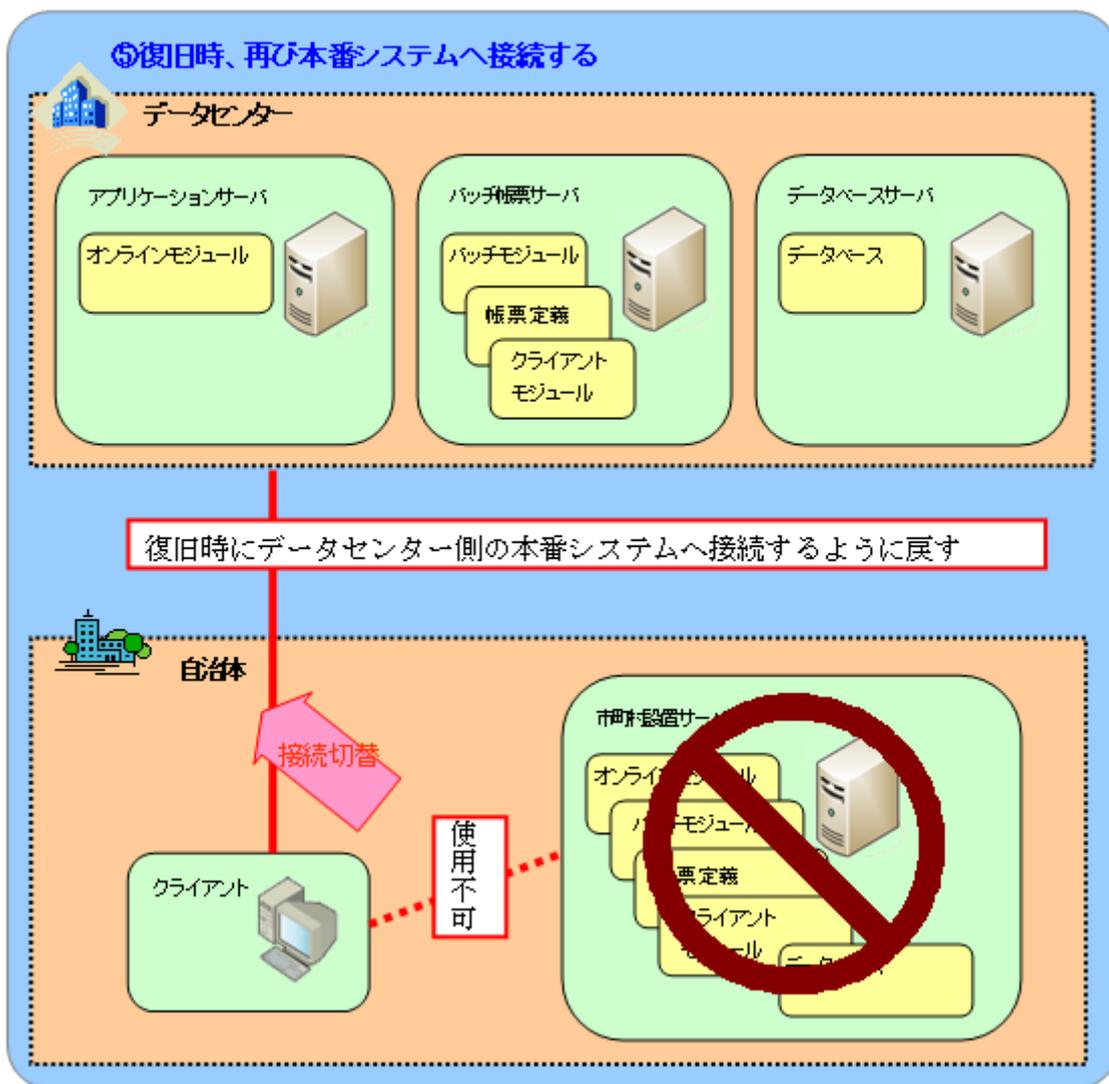


図 4-7 復旧時の本番システムへの切替

### (3) 実証の結果

実証結果は次のとおりである。

表 4-4 市町村バックアップのテスト結果

No	テスト項目	テスト結果
1	毎日、定時刻にデータセンター側のデータベースの内容が、縮退運転環境に反映される。	OK
2	毎日、定時刻にデータセンター側のモジュール等の内容が、縮退運転環境に反映される。	OK
3	通常（データセンター側が正常に動作している場合）、縮退運転環境では、システムが使用不可の状態にある。	OK
4	手動操作により、システムが使用不可の状態にある縮退運転環境を、使用可能な状態に切り替える事ができる。	OK
5	手動操作により、システムが使用可能な状態である縮退運転環境を、使用不可の状態に切り替える事ができる。	OK

下記①～⑤のように、確認結果は全て問題がなかった。

- ① タスクスケジュールに登録したバッチファイルの実行により、本番システムのデータベースをエクスポートしたダンプファイルを自治体設置サーバ側へコピーし、その後、縮退運転環境のデータベースにインポートを行った。インポートについては、特にエラーも発生せず問題なく処理が行われ、本番システムの全テーブルと縮退運転環境の全テーブルを比較した結果、各テーブルのデータ件数が一致することを確認した。
- ② タスクスケジュールに登録したバッチファイルの実行により、本番システムの「オンラインモジュール」、「バッチモジュール」、「帳票定義」、「クライアントモジュール」をセンター側から自治体設置サーバ側へコピーし、その後、縮退運転環境に配備した。コピー、及び配備については、エラーも発生せず問題なく行われ、本番システムと縮退運転環境の全ファイルが一致することを確認した。
- ③ 上記①、②の実証作業により縮退運転環境の構築が完了した。この状態では、まだクライアントマシンから、縮退運転環境へ接続してもシステムが起動しないことを確認した。なお、本番システムからデータベース及びモジュールのコピー、そして配備までの構築時間は約30分程度であった。
- ④ 縮退運転環境を使用可能にするためのバッチ処理を実行し、クライアントマシンから縮退運転環境へ接続してシステムの起動を行ない、問題なく縮退運転環境でシステムを使用できることが確認できた。なお、バッチ処理により縮退運転環境を起動可能な状態に切り替える時間は約1秒であり、業務が利用者が意識する事もない時間で行えた。
- ⑤ なお、正常にシステムが動作するかについては下記のテスト内容によって確認した。

表 4-5 縮退運転での動作確認結果

No	テスト項目	テスト結果
1	トップページにアクセスできること	OK
2	クライアントモジュールのバージョンが最新であること	OK
3	メンテナンス画面にログインできること	OK
4	システムにログインし、総合メニューが表示されること	OK
5	想定したデータベースサーバに接続されていること	OK
6	想定したアプリケーションサーバに接続されていること	OK
7	即時帳票が正しくプレビューできること	OK
8	バッチ処理(スプールあり)が正しく実行できること	OK
9	スプール帳票が閲覧できること	OK
10	端末使用状況が確認できること	OK
11	ヘルプファイルが表示できること	OK

- ⑥ 縮退運転環境を使用不可にするためのバッチ処理を実行し、クライアントマシンから縮退運転環境へ接続してシステムが再び使用できない状況に戻っていることを確認した。またデータセンター側に接続し、システムが使用できることを確認した。  
 なお、バッチ処理により、縮退運転環境を使用不可能な状態に切り替える時間は約1秒程度であった。

#### (4) 実証の考察

当実証実験により、タスクスケジュールによって指定した時間に、データセンター側で本番環境として動作しているシステムをそのまま自治体設置サーバ側にコピーし、エラー等問題もなく縮退運転環境を構築することが実証できた。その構築時間も約30分ということで、業務に支障がない深夜に実施する時間として問題ないといえる。また、縮退運転環境へのシステム切り替え時間、障害復旧時のデータセンターへのシステム切り替え時間、これらはどちらも約1秒であることから、運用的に問題のない動作時間であるといえる。

しかし、縮退運転環境は、本番システムのコピーであるため、本番システムと全く同一の機能を提供することが可能であるが、データセンター側で障害が発生した最新時点のコピーではない。縮退運転環境を構築する時刻は、本番システムの運用時間との折衝上、通常業務が終了後の深夜実行としてタスクスケジュールに登録されるのが最も一般的であるといえる。このため、例えば昼の12時に障害が発生して、データセンター側と自治体側の回線が不通になった場合、その日の朝から昼12時までの業務内容は、縮退運転環境に反映されていないことになる。

このため、「障害の発生時間」、「障害の復旧時間の目処」、「障害箇所」

を考慮して縮退運転環境での運用方法を変更し、対応する必要がある。以下にその内容を挙げてみる。

#### ア) 障害の発生時間による運用について

バックアップ時点から早い時間に障害が発生し、前日深夜のバックアップ時と縮退運転環境の内容について比較的差が少ない場合は、下記の運用手順が考えられる。

- ① 縮退運転環境に、朝～障害発生時点分の業務遂行内容を手作業で反映する。
- ② 縮退運転環境で、通常通り業務を遂行する。
- ③ データセンター側の本番システムが復旧後、本番システムのデータベースの内容を縮退運転環境のデータベースに置き換える。データベースは1自治体単位に管理しているため、置き換えによる他自治体への影響は無い。

逆に、バックアップ時点から遅い時間（例えば通常業務終了間際）に障害が発生し、前日深夜のバックアップ時と縮退運転環境との内容の差が多い場合は、下記の運用手順が考えられる。

- ① 朝～障害発生までに行った更新について調査する。
- ② 縮退運転環境で業務を遂行するが、システムの参照・更新の処理時には、上記①の調査分を考慮する必要がある。また更新の処理を行った場合は、処理内容を記録する。
- ③ データセンター側の本番システムが復旧後、上記②で記録した更新処理と同じ操作内容を本番システムに対して行い、同期をとる。

#### イ) 障害の復旧時間のめどによる運用について

障害の程度が軽く、復旧時間があまりかからないと見込まれる場合は、上記に記述した通りの内容で対応ができるが、復旧の見込みが立たない場合は、当分の間、縮退運転環境での運用が継続することを踏まえての対応が必要となる。

このため、上記の「遅い時間に障害が発生した時」のように本番復旧まで、縮退運転環境を仮環境として運用する（縮退運転環境上で行なった「更新を伴う操作」を記録しておき、復旧後に本番システムで記録通りの同じ操作を行って同期をとる）よりも、「早い時間に障害が発生した時」のように縮退運転環境のデータベースを本番システムとして扱えるように調整し、復旧後もそのまま縮退運転システムのデータベースを使用する（本番システムと差し替える）方法を取る判断も必要になるとと思われる。

#### ウ) 障害箇所による運用について

障害要因がネットワークではなく、モジュールに関連する箇所により限定される場合は、障害発生時点のデータベースを縮退運転環境にコピーして業務を遂行する事もできる。

しかしモジュール等のネットワーク以外の要因については、一般的に冗長化されているので、縮退運転環境を使用する必要もなくデータセンター内の冗長化の範囲で対応が可能である。やはり縮退運転環境を使用するような規模の障害は、データセンターの災害時に限られると思われる。

最後に、当初縮退運転環境での運用内容については、証明発行と台帳参照のみに限定し、縮退運転中の更新は行わなくても問題はないと考えていた。しかし、障害停止時間中の運用対応や、復旧時のデータベースの同期について考慮すると、上記ア) やイ) の方法のように、縮退運転での更新処理は必要(※)であると判断する。ただし、上記イ) に記述しているように障害復旧時間の目安が例えば30分以内に復旧する見込みがあるならば、更新が必要な処理は一旦保留(復旧するまで)しておき、更新処理を伴わない証明発行と台帳参照のみに限定して運用するという対応も一案である。

いずれの運用方法をとるかは導入自治体の規模や運用方法によるため、当実証実験で得られた内容をもとに検討して頂きたい。データセンターを利用することで庁舎内にサーバ機やデータを置かなくて済むとはいえ、市民へのサービスが停止することが無いよう、高可用性を実現するためには当実証実験で行ったような縮退運転の環境が必要であり、そして実現することも可能である。

- ※ 縮退運転環境のデータベースは前日の状態であるため、当日に処理されたデータに対しては処理できない。また、縮退運転中に台帳を更新する必要がある場合、更新を行わないとすれば、以後、そのデータについては処理できない。また、復旧が遅延した場合に復旧後のデータベースの同期化について負担やリスクが高くなる。

### 4.1.2 自治体クラウドコンピューティング

クラウドコンピューティングの特長として、物理サーバを意識しなくても運用が可能となる仮想化技術の活用が挙げられる。仮想化によるハードウェアの集約は、コストの削減、ハードウェアが故障した時の切替等による可用性の向上、必要な時に必要なリソースを確保すること（拡張性）が可能になった。

自治体クラウドにおいてもこれらを実現できることを確認した。

#### (1) 府と市町村の基盤共同利用

##### ア) 実証の概要・目的

##### A. 対象とする業務アプリケーション

仮想化技術を用いて共通の自治体クラウド基盤を構築し、次の2システムを自治体クラウド基盤上に構築した。このとき、既設システムで課題となっていたサーバ資源不足を解消する形でのシステム構築を行った。

表 4-6 本実証の対象業務アプリケーション

団体名	業務アプリケーション
京都府	文書事務支援システム
京都府自治体情報化推進協議会	共同利用型文書管理システム

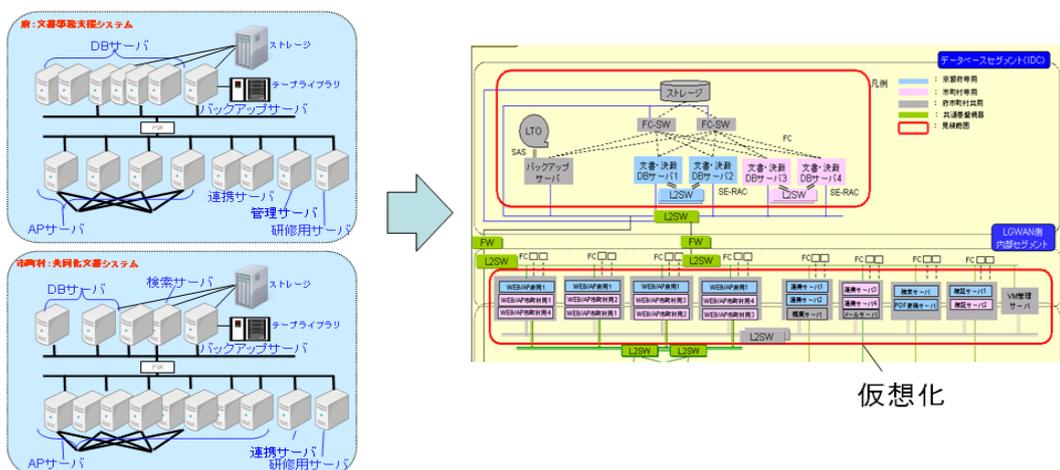


図 4-8 府と市町村の基盤統合イメージ

##### イ) 実施内容

府で利用している文書事務支援システムと、市町村で共同利用している文書管理システムを、クラウド基盤に集約させることで、サーバ台数を減

小させる。

#### A. 実施手順

京都府域データセンターに自治体クラウド基盤を構築する。

構築した自治体クラウド基盤に、2つの業務アプリケーション「府文書事務支援システム」「共同利用型文書管理システム」を構築し、ハードウェア統合を行う。

#### B. 具体的な内容

##### a. 自治体クラウド基盤の構築

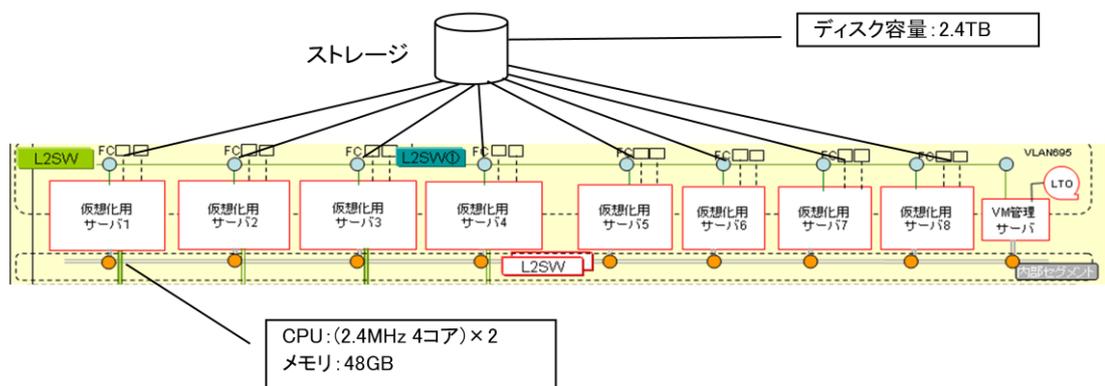
自治体クラウド基盤は大きく分けて、アプリケーション層とデータ層に分けて構築を行った。

#### [アプリケーション層]

仮想マシンを稼動させる仮想化用サーバ8台と、仮想環境を統合的に管理するため仮想化管理サーバ（VM管理サーバ）1台を配置した。

仮想化用サーバは、次の点を考慮し、ファイバチャネル接続により共有ストレージにアクセスできる構成とした。仮想環境におけるストレージ利用のメリットは次の点があげられる。

- ✓ 高速で大容量のディスクに仮想マシンファイルを置くことで、複数の仮想マシンを環境上で動作させることができる。
- ✓ 各仮想化用サーバから共通にアクセスできる共有ディスクとして機能するため、仮想マシンの障害切り替えを行うことができる。
- ✓ 仮想マシンファイルがストレージに集約されるため、バックアップがしやすい。



#### [データ層]

RDBMSを稼動させるデータベースサーバ4台と、バックアップを統

合的に行うためのバックアップサーバ1台を配置した。

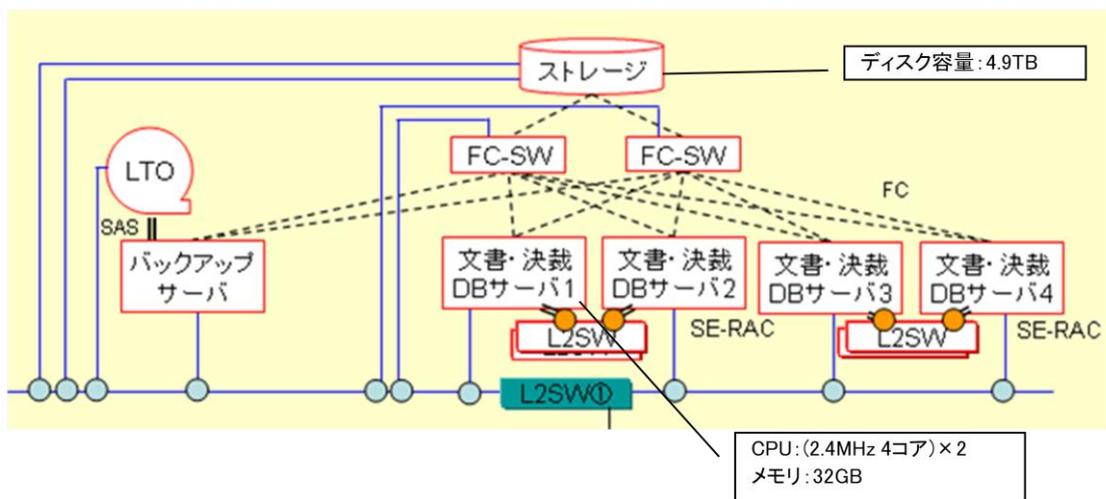


図 4-10 データ層の物理構成

b. 2つの業務アプリケーションの統合、仮想マシンの構築

仮想マシンは、テンプレートとなる仮想マシンを用意し、それらの複製（クローン）を利用して作成することで、構築作業の短縮を図った。

仮想マシンのリソース割り当ては、既設サーバのスペックをベースに下表のとおりとした。

また、業務アプリケーションの安定稼動を念頭に、仮想マシンがすべてのリソースを使用した場合でもリソース不足とならないように、仮想化用サーバ1台あたりの仮想マシン数は、最大4仮想マシンとした。

表 4-7 仮想マシン1台あたりの構成

項目	内容	備考
仮想CPU	2.4GH×2	
メモリ	WEB/AP：12GB その他：8GB	
仮想ディスク	C：30GB D：40GB	シン・プロビジョニング (仮想マシンで使用した分だけの領域を物理領域で確保)

さらに、「府文書事務支援システム」「共同利用型文書管理システム」のアプリケーション層のサーバは仮想マシンに移行した。各業務システムの仮想マシン数は下表のとおり。

表 4-8 府文書事務支援システムの仮想マシン

サーバ種別	仮想マシン数	用途
WEB/APサーバ	4	府文書事務支援システムの業務アプリケーション

サーバ種別	仮想マシン数	用途
		ョン本体
連携サーバ	2	連携機能の実行
PDF 変換サーバ	1	PDF 変換機能の実行
検索サーバ	1	キーワード検索機能の実行
検証用サーバ	1	動作検証用

表 4-9 市町村共同利用型文書管理システムの仮想マシン

サーバ種別	仮想マシン数	用途
WEB/AP サーバ	8	市町村共同利用文書管理システムの業務アプリケーション本体
連携サーバ	2	連携機能の実行
PDF 変換サーバ	1	PDF 変換機能の実行
検証用サーバ	1	動作検証用

表 4-10 業務アプリケーション共通の仮想マシン

サーバ種別	仮想マシン数	用途
帳票サーバ	1	帳票サービスの提供
メールサーバ	1	メールサービスの提供

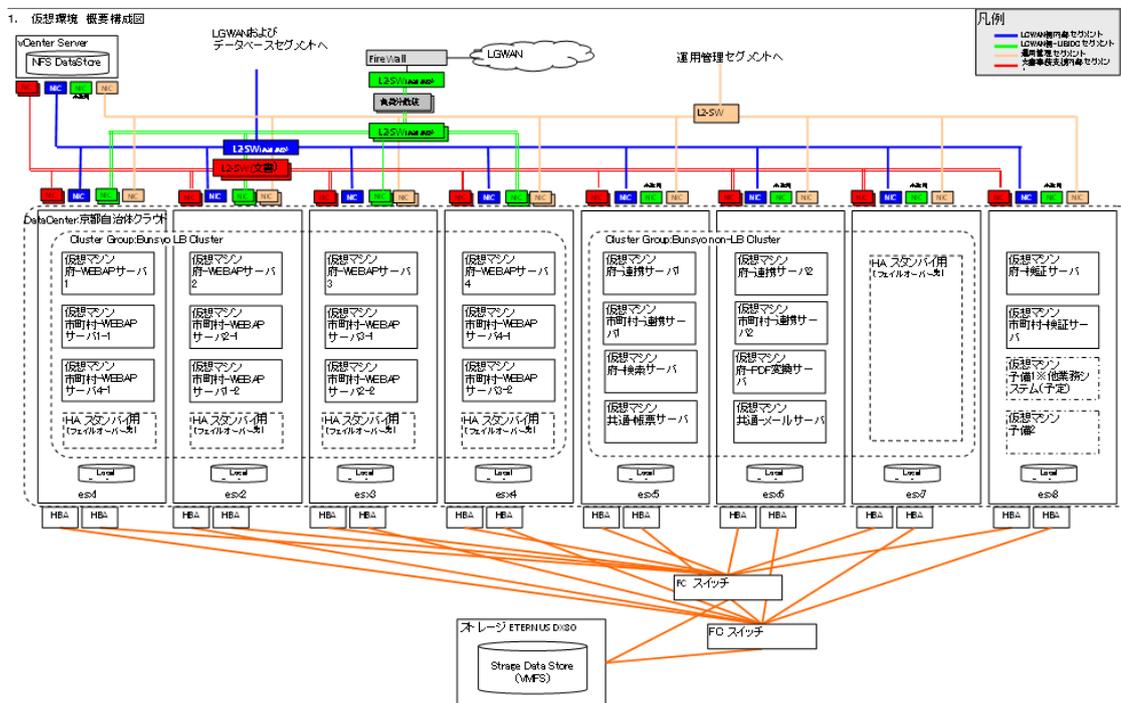


図 4-11 仮想環境の構成

## ウ) 結果

既設システムのサーバ台数は以下のとおりである。

表 4-11 既存システムのサーバ台数

業務システム	用途等	台数
京都府 文書事務支援システム	物理サーバ（AP、DB、連携、バックアップ等）	15台
京都府自治体情報化推進協議会 共同利用型文書管理システム	物理サーバ（AP、DB、連携、バックアップ等）	15台
合計		30台

なお、既設システムは利用率の増加にともない、サーバリソースの増強が必要な状況であった。既設システムのリソース不足を解消した構成でのサーバ台数は以下のとおりである。

表 4-12 仮想化せずに構築した場合のサーバ台数

業務システム	用途等	台数
京都府 文書事務支援システム	物理サーバ（AP、DB、連携、バックアップ等）	18台
京都府自治体情報化推進協議会 共同利用型文書管理システム	物理サーバ（AP、DB、連携、バックアップ等）	18台
合計		36台

今回の実証で構築したシステムでは、表2のサーバリソースを満足させる構成で構築を行った。このときのサーバ台数は以下のとおりである。

表 4-13 仮想化して構築した場合のサーバ台数

業務システム	用途等	台数
自治体クラウド基盤	仮想用サーバ (仮想マシン数合計：22台)	8台
	物理サーバ（仮想化管理、DB、バックアップ等）	7台
合計		15台

仮想化を活用した自治体クラウド基盤を適用することで、既設システムと比べてもサーバ台数を50%削減できることが実証できた。

$$1 - (\text{クラウド基盤台数 (15台)} \div \text{既設台数 (30台)}) \times 100\% = 50\% \text{ (削減率)}$$

さらに、自治体クラウド基盤上に構築したシステムは、サーバリソース増強後を想定しているため、「物理サーバで構築した場合の台数：36台」が、削減効果の比較対象の実態であり、削減率は58%となる。

$$1 - (\text{クラウド基盤台数 (15台)} \div \text{既設(増強想定)台数 (36台)}) \times 100\% = 58\% \text{ (削減率)}$$

このように当初想定していた削減率(30%)を大幅に上回る削減効果を得ることができた。仮想化により、サーバ削減効果が高いことが実証された。

### エ) 考察

この実証において、サーバ台数を58%以上削減させることに成功した。

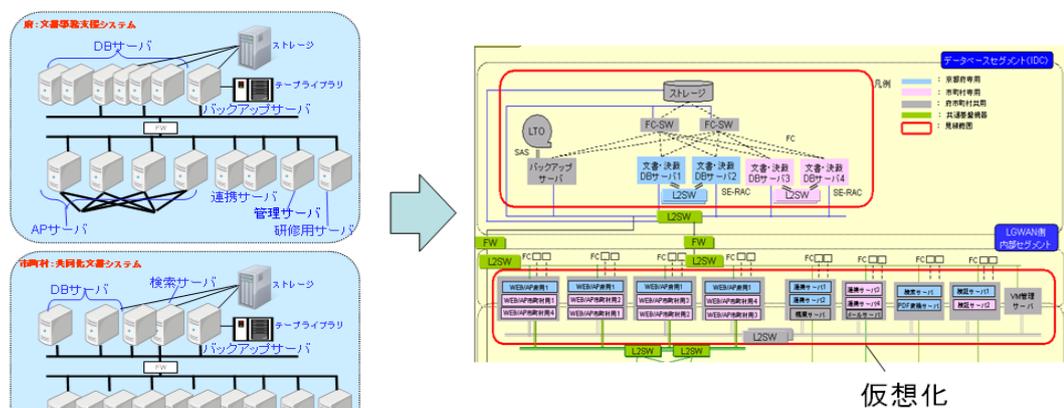


図 4-12 府と市町村の基盤統合イメージ

表 4-14 既設システムと自治体クラウド連携基盤のサーバ台数

業務システム	台数
＜府既設システム＞ 文書事務支援システム（※）	18台
＜市町村既設システム＞ 共同利用型文書管理システム（※）	18台
＜府・市町村基盤共同利用＞ 自治体クラウド連携基盤	15台

※ 既設システムのリソース不足を解消した構成

$$1 - (\text{クラウド基盤台数 (15台)} \div \text{既設 (増強想定) 台数 (36台)}) \times 100\% = 58\% \text{ (削減率)}$$

仮想化を活用した自治体クラウド連携基盤を適用することで、既設システムと比べてサーバ台数を58%削減できることが実証できた。

このように当初想定していた削減率（30%）を大幅に上回る削減効果を得ることができた。仮想化により、サーバ削減効果が高いことが実証された。

## 4.2 データセンター間接続実証

京都府データセンターにあるバックアップデータを北海道データセンターにバックアップすることで、万一京都府データセンターが災害等により利用できなくなった場合にも、データを保護し、復旧することができることを確認した。

データセンター間接続実証では「8.2 L GWANに関する課題」に報告したようにネットワークの通信障害が発生したところであるが、問題の発生から約一カ月が経過した11月中旬に、L GWAN接続装置の設定変更（FWの設定変更、ソフトウェアのバージョンアップ）を行ったところ接続自体は安定した。

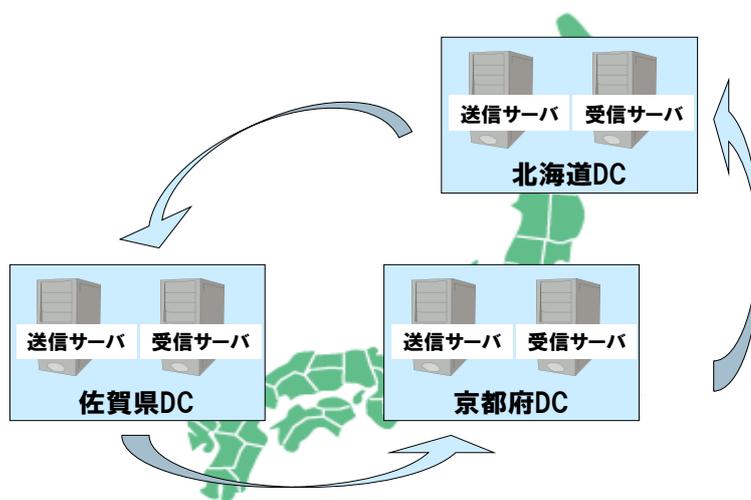


図 4-13 バックアップイメージ

### 4.2.1 データセンター間バックアップ

#### (1) 実証の概要・目的

##### ア) 実証の目的

従来のICT環境においては、広域災害に対応するためのバックアップサイトの構築は、地方公共団体ごとに準備する必要があったため、コスト面での負担が大きく、実現が見送られるケースが多かった。

自治体クラウドの環境で、都道府県域データセンターに相互にバックアップサイトを設けることで、広域災害に対応することができる。

本実証では、京都府域データセンターが広域災害により、壊滅的打撃を受けた場合を想定し、遠隔地へのバックアップを行うことで自治体において管理している業務データの消失を回避することができ、参加自治体及び住民の不安感を軽減できることを確認した。

イ) 実証の概要

北海道データセンターを京都府データセンターのバックアップサイトとして位置づけ、バックアップデータを、LGWANを経由して連携することで、バックアップを行った。また、バックアップしたデータを京都府データセンターのデータベースに復元することで、業務復旧が行えることを確認した。

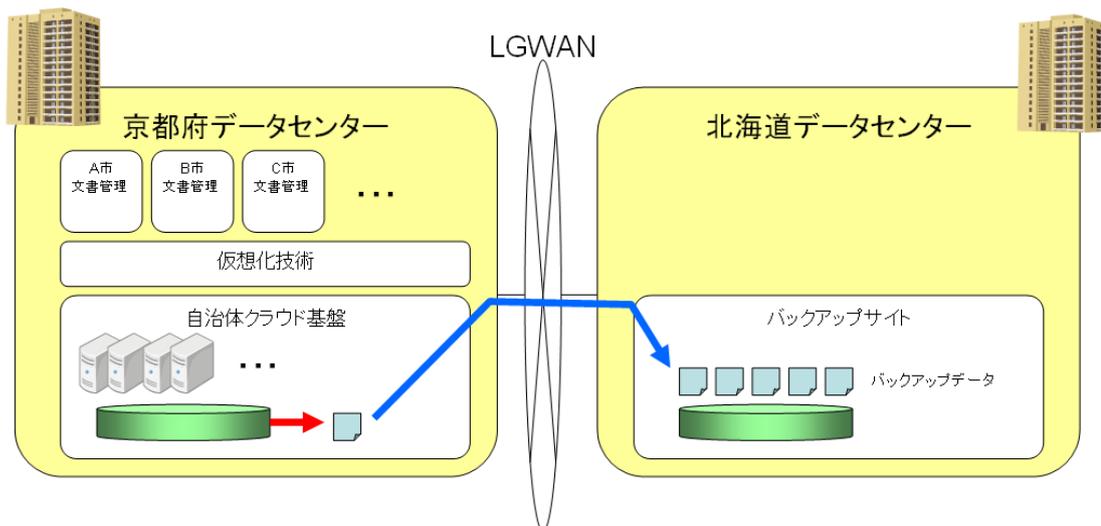


図 4-14 データセンター間バックアップのイメージ

(2) 実証の内容

ア) バックアップサイト

バックアップは京都府データセンターのデータを北海道データセンターにバックアップすることで実施した。

イ) データセンター間の接続方式

京都府データセンターと北海道データセンターはVPNによるトンネリングでの接続方式とした。IPSecによりネットワーク的にセキュリティを維持しつつ、バックアップに必要なプロトコル（ファイル転送用プロトコル）を利用可能とした。

ウ) バックアップデータ

バックアップデータは、京都府文書事務支援システムのデータベースのデータを使用し、業務アプリケーションを利用した場合の運用実体に近い形でのバックアップを行った。

表 4-15 バックアップデータ

種別	容量※	備考
フルバックアップ	2.4TB	

差分バックアップ	6.0GB	一日の実運用で発生する差分データ量
----------	-------	-------------------

※容量は非圧縮時の値

エ) ファイル転送プロトコル

バックアップサイトを本運用する場合、複数の都道府県データセンターで共有して利用されることが想定され、また、バックアップサイトのデータ送受信サーバで採用されるOSはWindowsやLinuxなど、様々な種類が想定される。ファイル転送にあたり、より多くのOSで標準的に使用できるプロトコルを使用することで、バックアップに係るソフトウェア依存性をできる限り小さくすることが可能と考え、ファイル転送の標準的なプロトコルであるFTPプロトコルを使用した。

オ) バックアップ方式

LGWANは利用できる帯域が8~10Mbps（理論値）と狭く、また、業務運用で使用されているネットワークであり、全てのバックアップデータをネットワーク経由で転送することは、LGWANの帯域圧迫によるネットワーク遅延などのLGWANを利用した業務への影響や、実効速度を3Mbpsとしたとき、ファイル転送時間が1ヶ月を超える試算となるなど、実用的でないことが想定された。

そこで、データ容量の大きいフルバックアップはテープにバックアップし、媒体を北海道データセンターに別送する方式とした。データ容量の小さい日々発生する差分データをバックアップファイルとしてファイル化し、LGWAN経由で北海道データセンターにファイル転送する方式とした。

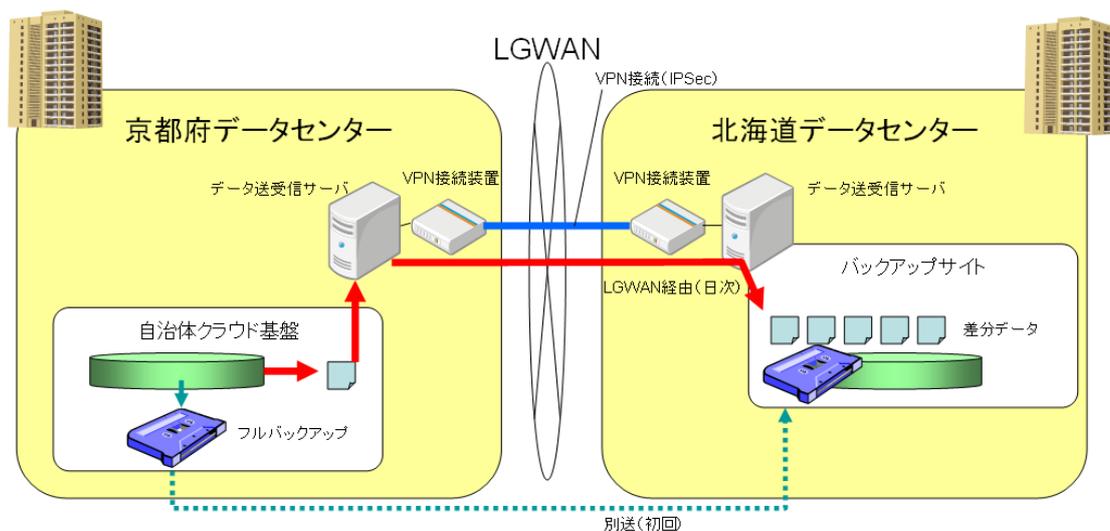


図 4-15 オフサイトバックアップの方式

#### カ) ファイル転送時の帯域制御

業務運用で使用されているネットワークであり、バックアップで帯域を圧迫することで、一般業務に影響がでるリスクがあった。そこで、バックアップに利用するプロトコルについて、Qos (Quality of Service) による帯域制御を行い、使用できる最大帯域(スロットル値)を48KBpsとした。

#### キ) リストア・リカバリ方式

L GWANは利用できる帯域が8~10Mbps(理論値)と狭く、また、業務運用で使用されているネットワークであり、バックアップ時と同様に全てのデータをネットワーク経由で転送することは実用的ではない。

そこで、データ容量の大きいフルバックアップのリストアは、北海道データセンターで保存する媒体で別送し、京都府データセンターでリストアする方式とした。また、差分データも復元時はフルバックアップと一体で取り扱うため、フルバックアップを搬送するときに併せて差分データについても搬送する方式が実用的である。このように、実運用を想定し、復元時はバックアップデータを媒体で輸送する方式とした。

#### ク) 実証で利用したツール等

本実証では次の機器、ソフトウェア及びツールを使用した。

表 4-16 実証に使用した機器

種別	用途
仮想化用サーバ	仮想環境を提供する物理サーバ
データベースサーバ	データベースのサービス提供
ストレージ	仮想環境の物理ファイルの保存 データベースのデータ保存(共有ディスク)
VPN 接続装置	京都府データセンター - 北海道データセンター間 をVPN 接続するための機器
データ送受信サーバ	バックアップデータの送信(京都側) バックアップデータの受信・保存(北海道側)

表 4-17 実証に使用したソフトウェア、ツール

種別	名称
仮想化用ソフト	VMware sphere 4.0 update 1(ESXi4.1)

種別	名称
OS	Windows 2003 R2 Standard Edition (64bit)
RDBMS	Oracle 10g R2
アプリケーションサーバ	Oracle Application Server 10g R1
業務アプリケーション	京都府文書事務支援システム

#### ケ) 実施手順

データセンター間バックアップは、次の手順で実証をおこなった。

- ① 文書管理システムの業務データ全体をバックアップし、テープへ保存する。
- ② 保存したテープを北海道域データセンターへ搬送する。
- ③ ①以降のデータを日次差分バックアップとしてバックアップする。
- ④ ③でバックアップした差分バックアップを京都府バックアップサーバ領域へ転送する。
- ⑤ ④で転送された差分バックアップを、北海道バックアップサーバ領域へ転送する。
- ⑥ 北海道域データセンターへ搬送したバックアップテープを京都府域データセンターへ搬送する。
- ⑦ 北海道バックアップサーバ領域で保存されている差分バックアップを京都府バックアップサーバ領域へ転送する。
- ⑧ ⑥で搬送したテープのバックアップデータを復元する。
- ⑨ ⑦で転送した差分バックアップを復元する。
- ⑩ 復元されたデータが正常であるか（異常がないか）確認する。

#### (3) 実証の結果

##### ア) フルバックアップの処理時間

種別	処理時間	備考
バックアップ時間 (Disc-to-Disc)	3 秒	オンラインバックアップ開始～終了までの時間
バックアップ時間 (Disc-to-Tape)	14 時間	運用と切り離して、テープへのバックアップの開始～終了までの時間

## イ) 差分データの転送

種別	計測値	備考
転送データ量	1.5GB	転送データ量 ※差分ファイル：6.0GB を圧縮 ※10MB 単位にファイルを分割して転送
データ転送時間	567 分	L GWAN経由でバックアップサイトにバックアップデータを転送する時間 途中でネットワークの切断があったため、次の単位で分割しての転送 (82 ファイル×10MB：294 分) (74 ファイル×10MB：265 分) (2 ファイル×10MB：7 分)
転送速度	46.5KBps	QoS で 48KBps に帯域制御した状況下での結果

## ウ) データのリストア（復元）の処理時間

種別	処理時間	備考
リストア時間（Tape-to-Disc）	14 時間	テープメディアからバックアップ領域のデータにコピーする時間
リストア時間（Disc-to-Disc）	4.1 時間	復元されたバックアップ領域のデータを業務領域にコピーする時間
リカバリ時間	16 分	データベースの整合性を保った状態までリカバリ処理を行う時間

## (4) 結果の考察

## ア) L GWANを利用したバックアップについて

日次の差分バックアップについて、L GWANの帯域を考慮した48KBpsであっても一日の差分データの転送が約10時間で完了できることが確認できた。これは、日次の差分バックアップが12時間以内で完了しており、L GWAN経由でも十分に運用可能であることが示された。

実証実施時点では、L GWANの機器設定が最適化されておらず、ネットワークが不安定であり、連続したデータの転送が難しい状況であったが、ファイルを分割することにより、既に転送済みのデータを再送することなく、途中から転送を開始できるようにする、帯域を絞りネットワーク負荷を下げるなどの施策により、そのようなネットワーク環境下でも、データセンター間のバックアップが可能であることが示された。

ただし、現状のL GWANは帯域が狭く、複数のサイトや複数の業務アプリケーションが互いに遠隔地にバックアップを行うことを考えると、ネットワーク帯域が十分に確保される必要があると考える。現状のL GWANのネットワーク帯域は十分であるとは言えない。今後、L GWANの帯

域を拡張していくことが望ましい。

イ) 業務復旧の即時性について

バックアップを遠隔地に保存した場合、有事の際のデータ復旧には次のステップが必要となる。

- ① 復元先のハードウェアの復旧
- ② 遠隔地からのバックアップデータの搬送
- ③ バックアップデータのリストア・リカバリ

災害時の被害の状況によるが、ハードウェアが全壊した場合、代替機の準備、ハード再構築など数週間にわたる作業が必要となり、その間、業務が行えない状況が想定される。

有事の際に、業務復旧の即時性をあげるには、データ同期など、バックアップサイトで即座に業務データが利用できる状況にあることが望ましい。

しかしながら、バックアップサイトで業務データを利用できる状況とするには、運用側のサイトと類似の機器、ソフトウェアをバックアップサイトにも配置する必要があり、コスト面での課題がある。運用側に依存しない異種間での同期などの技術の導入が期待される。

## 4.2.2 データセンター間接続追加実証

北海道データセンターを京都府データセンターのバックアップサイトとして位置づけ、バックアップデータのリアルタイムバックアップを行った。また、大規模障害を想定し、バックアップサイトでの業務復旧が行えることを確認した。

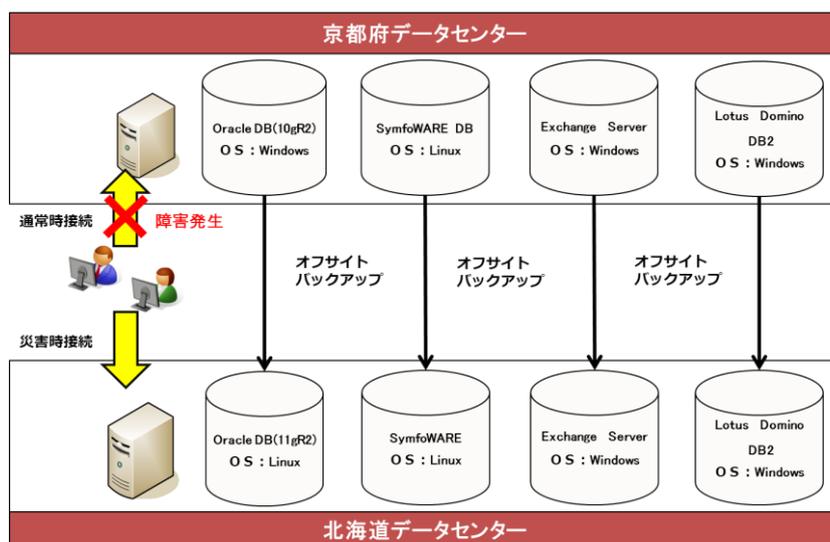


図 4-16 データセンター間接続追加実証イメージ図

データセンター間バックアップの実証によって、様々なデータベースの特長や性能について確認した。

### (1) 行政機能のバックアップと復旧（異なるOS、異なるバージョンのデータベースのバックアップ）

オフサイトバックアップに係る本実証においては、行政システムのバックアップを遠隔地に設置したバックアップ拠点（拠点間はL GWANにて相互接続）に対し行えるか、という視点を中心に置き、クラウド技術を活用した効率的かつ実用に耐え得る行政機能バックアップシステムの実現性検証を行った。

また、今回の検証では、実用性・実現性を意識した観点から、京都府において実際に行政システムとして利用されている「京都府文書事務支援システム」のアプリケーションや業務データを利用した。

以下に今回のオフサイトバックアップ検証に準備した項目とそれぞれの概要および目的を記す。

#### ア) 行政機能データバックアップ

異なる環境へのバックアップサイトへのデータ転送（通常運用時）

### A. 実証の概要・目的

通常運用時においては、常時バックアップサイト（北海道）へデータを転送することにより、業務本番運用に必要なデータが安全なバックアップサイトに常時保存されている状況を確認した。

また、このオフサイトへのデータ転送実験の前提として、異種システム環境が混在している状況を前提とし、京都府環境と北海道環境で異なるシステム環境（OS, 異なるバージョン）を用意の上検証を行う。これにより単なるオフサイトへのデータ転送の検証に加えて、多対1（N：1）のバックアップシステムの実現可能性を合わせて検証することができる。

クラウド型のシステム普及後には今まで以上に重要になることが想定される、遠隔地へのデータの同期保存の実用性を確認すること、また、システム毎に異なる環境を前提とした多対1のバックアップの仕組みを検証することで、より経済効率が良いオフサイトバックアップの実現性を確認することを、本検証の目的と設定する。

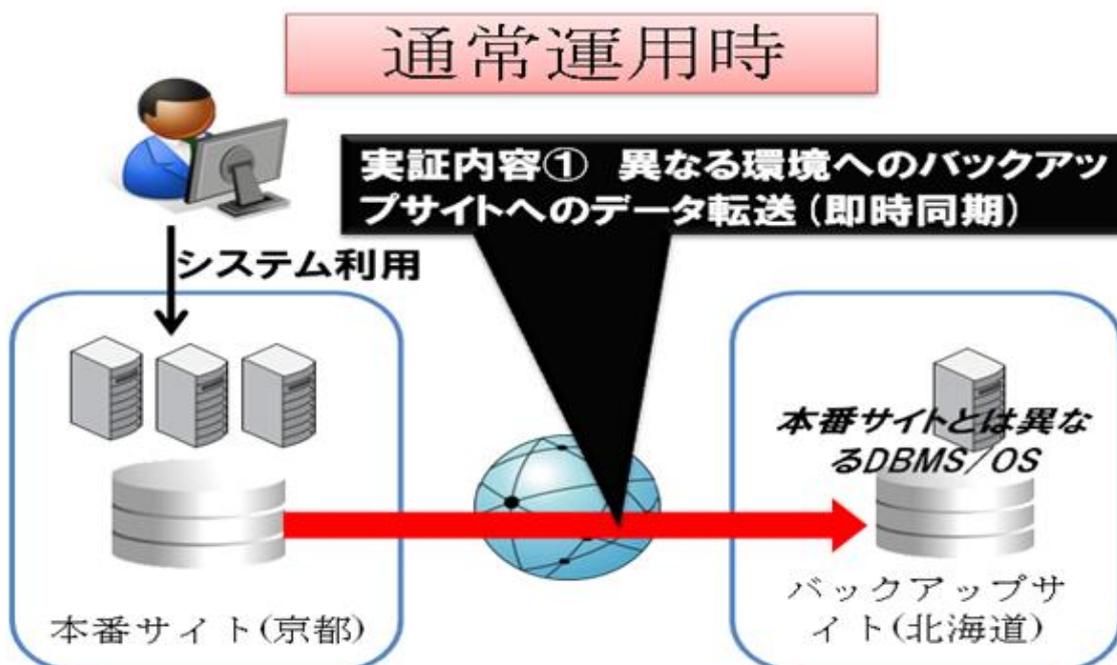


図 4-17 異なる環境へのバックアップサイトへのデータ転送（即時同期）

## B. 実証の内容

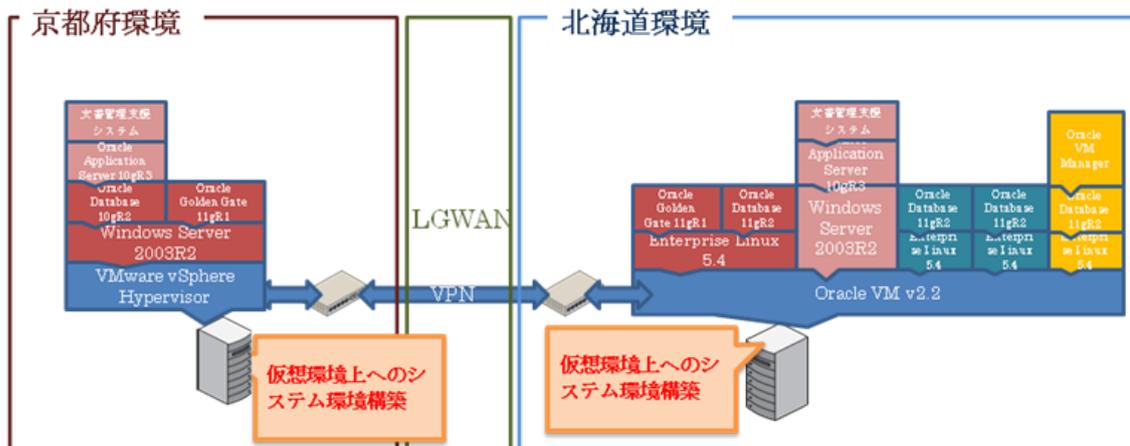


図 4-18 実証環境

### a. 基盤環境構築

本検証では、バックアップサイトも含めてクラウド化された環境を構築することを前提に、全てを仮想化しクラウドに見立てた環境を用意し、構築を行った。

事前に仮想化環境上に作成されたシステムを持ち込み、これを導入した。

京都府・北海道双方において、仮想化環境を導入しこの環境上に仮想化されたアプリケーション環境および、バックアップ・データベース環境の構築を行った。

表 4-18 京都府と北海道の環境

	京都府	北海道
仮想化環境	VMWare vSphere Hypervisor(ESXi) 4 (300MB)	Oracle VM 2.2 (398MB, 580MB)
オペレーティングシステム環境(仮想環境上)	Windows Server 2003R2	Windows Server 2003R2(京都府文書事務支援システム, Enterprise Linux 5.3(データベース))
仮想イメージ	京都府文書事務支援システム (含 Oracle Internet Application Server 10gR3, Oracle Database 10gR2) 40GB *別途 Golden Gateを導入	1. Oracle VM 管理サーバ - 580MB 2. 受信用データベースOracle Database 11gR2, Golden Gate 11gR1) - 3.2GB 3. 拡張用データベース・サーバ(Oracle Database 11gR2, Real Application Clusters 11gR2)- 4.3GB 4. 京都府文書事務支援システム(含Oracle Internet Application Server 10gR3, Oracle Database 10gR2)- 40GB

「京都府文書事務支援システム」の実稼働環境を仮想化して、京都府・北海道に用意した。これはデータベース環境を業務データとして稼働させ、連携することでバックアップ実証を行うための環境となった。

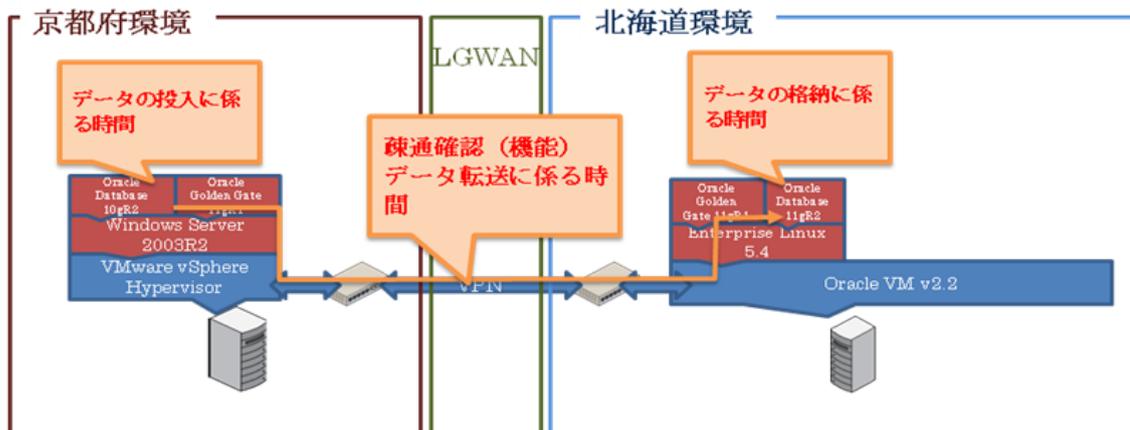


図 4-19 バックアップ実証概要

b. 検証環境

京都府環境の「京都府文書事務支援システム」のデータベースから、北海道環境の受信用データベースに対して、データを転送することでオフサイトバックアップを実現する。転送機能としてGolden Gateを京都府、北海道に導入して実装する。

行政機能「京都府文書事務支援システム」で利用されているデータを元

に、データバックアップを北海道に転送する、実証を行う。ここでは、データベース全体を一括してあらかじめ同期しておき逐次更新される行政データを視野に、バックアップを一件毎に行うことにより災害発生時に、発生直前までのデータがバックアップサイトに反映することができるかを確認した。

1. 京都府文書事務支援システムデータベースと転送機能の接続設定
  - I. 転送機能（Golden Gate）からデータベースとして稼働しているOracle Databaseへの接続を実施（TCP, Oracle通信）
  - II. 転送機能（Golden Gate）の管理コンソールログより、データベースへの接続確立を確認
2. 北海道、受信用データ転送機能との通信疎通の確認
  - I. LGWANを通じて、京都府・北海道サイトの転送機能（Golden Gate）の相互通信を設定
  - II. 転送機能（Golden Gate）の管理コンソールより、相互接続確立確認（接続履歴より確認）
3. 一部の業務データを用いた基本的な転送の確認
  - I. 「京都府文書事務支援システム」のデータベースから、一部を抽出した表を使用
  - II. SQL文を使用して、このテーブルに新規の行を投入（1件600byte）
  - III. 京都府側で、投入データの確認（SQLを使用して、確認）
  - IV. 北海道側で、投入データの確認（SQLを使用して、確認）
  - V. 転送機能（Golden Gate）の管理コンソールから、履歴を確認
    - \*いずれも同数のデータが投入されているか確認
4. 一部の業務データを用いて、件数を増やした転送の確認
  - I. 3の検証と同じ、「京都府文書事務支援システム」のデータベースから、一部を抽出した表を使用
  - II. SQL文を使用して、一件からはじめ、データを連続して投入し検証を実施
  - III. 京都府側で、10件、100件、1000件、5000件、10000件、50000件、100000件のデータを用意し、順次表にデータを投入する。
  - IV. 京都府側で、投入データの確認（SQLを使用して、確認）
  - V. 北海道側で、投入データの確認（SQLを使用して、確認）
  - VI. 転送機能（Golden Gate）の管理コンソールから、履歴を確認する。
    - \*いずれも同数のデータが投入されているか確認

## C. 実証の結果

## a. 総評

クラウド環境をモチーフとした仮想環境については、速やかに環境構築ができた。

データ転送件数を段階的（1件～100,000件）に増やし、L GW ANを介したオフサイトバックアップサイトへの転送が行えることが確認できた。またデータ欠損等なく、データの信頼性の確認ができた。

## b. 結果詳細

## A) 基盤環境構築について

実証を行うにあたっては、クラウド環境をモチーフとした仮想化環境を構築した。この仮想化環境の中には実証に用いるOS環境、データベース環境ならびにデータ転送機能を含めた、管理コンソール、アプリケーションを用意している。環境構築にあたっては、およそ3時間で仮想環境（H y p e r v i s o r 環境）から、仮想サーバ・イメージの導入を完了することができた。

表 4-19 検証結果

## 京都府環境

	テスト環境での時間	検証結果
Vmware vSphere Hypervisor(仮想化環境) 導入(300MB)	30分	30分
仮想イメージ導入 (京都府文書事務支援システム 40GB)	4時間	2時間
転送機能(Golden Gate) 導入、動作確認	2時間	1.5時間

## 北海道環境

	テスト環境での時間	検証結果
Oracle VM(仮想化環境) 導入 (398MB, 580MB)	30分	30分
仮想イメージ導入 (受信DB 3.2GB, 拡張用DB 4.3GB, 京都府文書事務支援システム 40GB)	6時間	3時間

転送機能（Golden Gate）導入、動作確認	（仮想サーバ導入に含む）	（仮想サーバ導入に含む）
--------------------------	--------------	--------------

北海道での基盤構築については3時間で完了した。自社内テスト環境での構築では6.5時間かかり、この結果を当初の想定時間としていたが、大幅に短縮することができた。

環境構築後、実データとして、「京都府文書事務支援システム」の一部のデータを用い疎通確認および単純なデータ転送について、転送機能ミドルウェアの内部コマンドによる動作確認を行うことができた。一件あたり600byteのデータを、1秒以下での転送・適用を確認している。

## B) 業務データバックアップの検証について

「京都府文書事務支援システム」の業務データバックアップに主眼をおき検証を実施した。主に、一括のバックアップ（100,000件規模）に向け、段階的に件数を増やすことにより実証を行う予定で実施していた。

しかし作業開始直後から、VPNが不安定なことから回線切断が生じ、データ転送を予定変更し、環境構築時点で実証された少量のデータ（1件/600byte）から始め、段階的に件数を引き上げることで行うこととした。

結果として、一度の転送においては5,000件データ（およそ3MB）あたりおよそ7sで転送、北海道におけるデータ適用については10sで反映することができた。10,000件（6MB）転送においては転送を実施したところ、途中転送が停止した。（転送履歴より確認）状況としてVPNが回線切断されており、転送が停止されたもので状況確認したところ京都府データベースでは10,000件のデータは正しく投入されていた。VPN回線復旧後、北海道データベースを確認したところ9,964件（データ量として、約5.3Mbyteに相当）が確認されたが36件は転送されていなかったことが検証された。

なお、出力データの結果についてはSQLのSELECT文を用いて京都府・北海道双方の表を参照し、目視でのデータ比較を行い同一のものであったことを確認している。

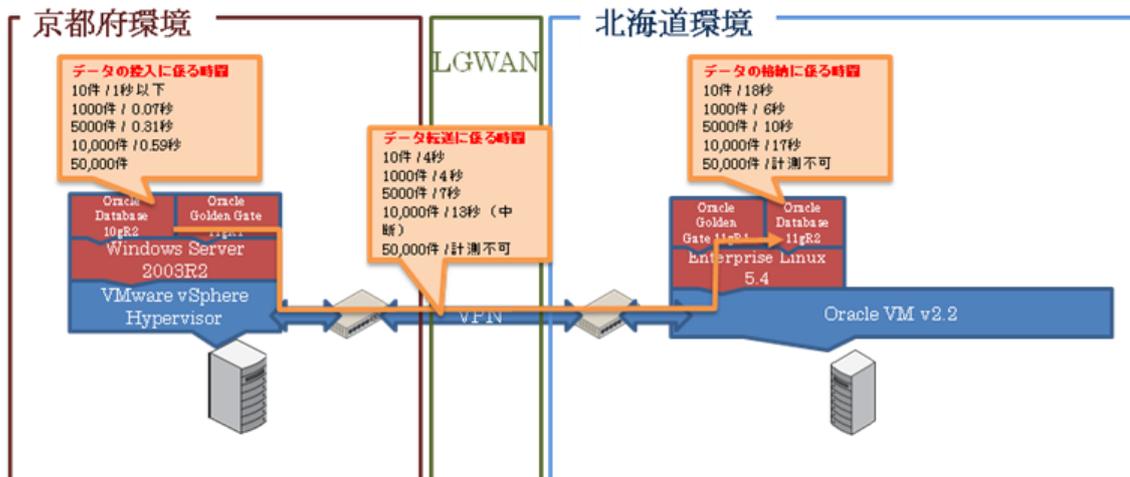


図 4-20 実証概要図・および結果概要

表 4-20 京都府文書管理システム・本番テーブルを用いた検証の実施結果（詳細）

No	京都環境								北海道環境				
	INSERT 件数	コミット 間隔	CPU 使用率(%)	更新データ 生成量(Byte)	更新処理 時間(s)	更新取 得まで の時間 (s)	送信完 了まで の時間 (s)	送信N/W負 荷(Byte/s)	受信N/W負 荷(Byte/s)	INSE R T 件数	CPU使用 率(%)	適用ま での時 間(s)	
1	10	1	1.70%	4,882	(1秒以	1	4	1,393.36	611.91	10	50.00%	18	
2	1,000	1	2.10%	499,914	0.07	1	4	52,356.14	53,116.98	1,000	50.00%	6	
3	5,000	1	5.70%	2,501,998	0.31	1	7	251,443.72	265,394.90	5,000	49.00%	10	
4	10,000	1	10.54%	5,003,996	0.59	1	13	331,199.31	372,455.79	9,964	50.00%	17	
5	50,000	100	32.42%	23,761,944	10.08	1	-	送信できず	受信できず	0	-	-	
※	0	0	0.3~1.7%	-	-	-	-	500~1,800	85~120	0	0~2	-	

※同期実行時以外(何も実行していない時の状態)

1件あたりおよそ600byte

## D. 結果の考察

### a. 基盤構築について

当初の想定時間（テスト環境での構築時間）を3.5時間短縮することができた。

自社内での事前検証では、仮想サーバ・イメージ導入（コピー作業）が、6時間かかっており、全く同じ仮想サーバ・イメージのコピーを行うための検証を想定した。

しかし実証機では、同じのデータサイズであるにも関わらず大幅に時間短縮がなされた。この結果から、ディスク性能が特に高いことで当初想定した時間より早く終了したものと考えられる。

### b. 都道府県間のバックアップについて

LGWANを介して、京都府から北海道へのオフサイトバックアップの実施に、今回成功することができた。

転送機能を用いた疎通確認（管理コンソール、接続履歴の参照）により

ネットワーク上での正常な接続が確認されたことから、通信上の制約はないことが確認された。

600byte/1件、5Mbyte/10,000件の業務データを用いた検証でも、京都府データベースで投入されたデータが、北海道データベースでも反映されたことからL2WANを介した、都道府県間のバックアップについて、問題ないものと判断した。

#### c. 異なるOS間のバックアップについて

京都府環境では、Windowsを用いた環境を用意し、北海道ではLinuxを用いたOS環境で構築した。

業務データ転送に関わる部分では、特にOSに起因した障害や問題は特に生じていない。

これら結果から、異なるOS間であっても問題なくバックアップ環境が構築できるものと判断した。

#### d. 異なるバージョン間のバックアップについて

京都府環境では、Oracle Database 10gR2を用いたデータベース環境、北海道環境ではOracle Database 11gR2を用いたデータベース環境を用意した。この検証では2つ上のバージョンのデータベースへバックアップを行ったが、異なるバージョンに起因した障害や問題は特に生じなかった。

また、後述するデータ品質についても問題はなく、異なるバージョン間であってもバックアップ環境が構築できるものと判断した。

#### e. データの品質について

京都府環境で投入された業務データは、転送先の北海道環境データベース表にSQL（SELECT文）を実行、結果を出力し京都・北海道の結果を目視で比較したところ、データの品質（正しいデータになっているか、データの欠損等ないかレコード数の確認）に関して問題はなかった。このことから今回検証した統合的なバックアップシステム方式は、自治体クラウドにおいて適用可能であると判断した。

#### f. ネットワーク回線の品質について

転送件数を徐々に増加させたデータ転送においては、10,000件以上のデータ転送が不成功に終わったが、これはL2WAN上に構築されたVPNに問題が生じ、ネットワークが切断されたことが原因であることがわかっている。その他のデータベース、サーバ、バックアップソフトなどには異常は認められなかった。

なお検証の当日、VPNの動作が不安定な状況であり、また転送機能（G

olden Gate) の圧縮効率が正確に測れていないため10,000件および5.3Mbyteのデータ量がネットワークの限界値かどうかを判断することは出来ない。

g. 大規模データへの対応について

今回の検証では、データ転送量を抑えるために、更新データ（トランザクション単位のデータ）を基本としたデータ転送を行う方式を採用し、また更新データについても圧縮する機能も実装している。これら機能を活用しつつ、バックアップシステム全体の最適なデータ転送量を割り出した上で、ネットワーク負荷をあまりかけない形態での統合的なオフサイトバックアップシステムの現実的な実装を行うことが可能であると考察する。

イ) 災害対策サイト（iDC）におけるデータベース・サーバ拡張  
バックアップサイトにおけるスケールアウト（障害発生時）

A. 実証の概要・目的

バックアップサイト（北海道）に仮想化環境を用意し、本番サイトでの障害発生時のバックアップサイト側における機能復旧時にシステムリソースの拡張（スケールアウト）を柔軟に行えるか検証した。これは、平常運用時においてバックアップシステムはデータバックアップを行うことが出来る最低限の台数（データベース・サーバ）で稼動しつつ、業務切り替え時に柔軟に必要な規模でのシステム拡張が出来るかを確認した。

バックアップシステムの構築・運用に係る費用の最小化とバックアップサイトの運用の効率性向上を図る実証となった。

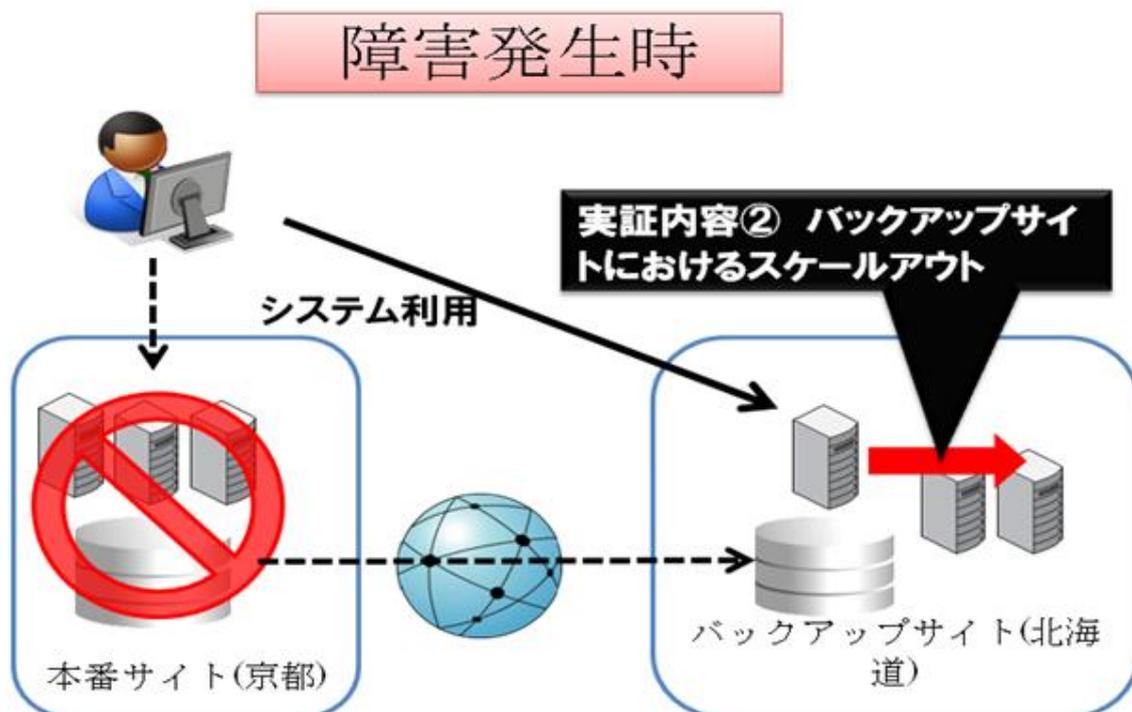


図 4-21 バックアップサイトにおけるスケールアウト

B. 実証の内容

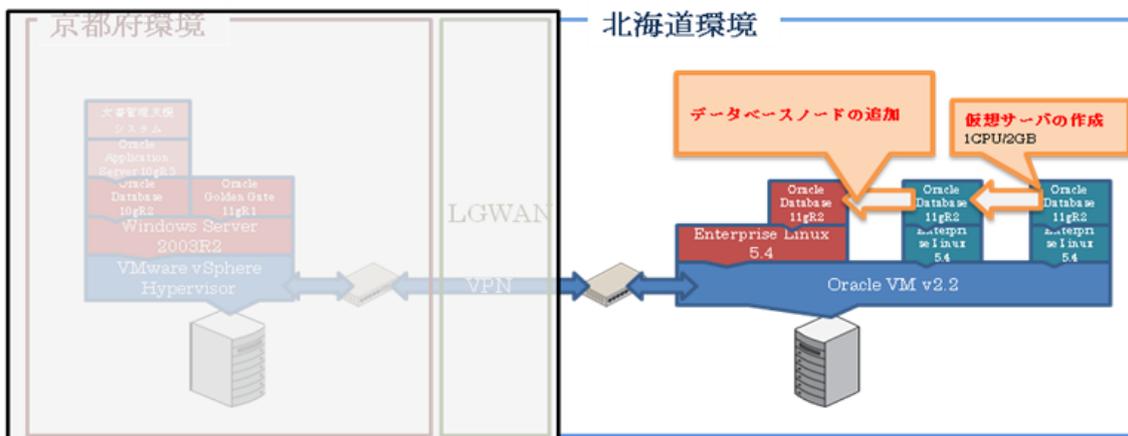


図 4-22 実証内容イメージ

本実証では、バックアップおよび業務アプリケーション接続・復旧に用いたデータベースを、仮想化を用いたクラウド環境から作り出したサーバを増設することにより、拡張を行った。

拡張にあたっては、サーバを複数並列化させることで性能を向上させることができるデータベース・クラスタ拡張機能 (Real Application Clusters) を用いた。この機能では、ネットワークを介してデータベース情報をサーバ間で通信しあう機構を用いて並列化を行い、稼働しているサーバ台数分の性能を出すことができ、データベース・クラスタ

内のサーバがダウンしてしまった場合であっても残ったサーバで稼働しつづけることができる。

この検証では、平常時最低限で動作しているバックアップ・データベースでは災害発生時、性能を満足できないと考えられることから拡張性を検証することを目的とし、必要時に応じてサーバ資源を借り受けることができるクラウド技術を利用することにより、有事の際に行政機能を十分にこなすことができるデータベース性能を確保することができるか、および運用コストを抑制することができるか検証をする。

サーバ資源については、1サーバしかなくかつ性能が限られるためデータベースに絞って実施した。ただ、データベースについてはバックアップ検証に用いたデータベースと同様のものを利用しているため、実証において問題ないものと確認した。

#### a. 検証環境

北海道環境において、受信用データベース・サーバに、新たに作成されたデータベース・サーバ（仮想環境から新たに創り出した仮想サーバ）を追加して、データベース・クラスタ化を行うことで拡張を行い、1台から3台分の環境に拡張する検証を行う。

データベース・クラスタ化に当たっては追加する毎に性能を検証し性能が向上しているかを確認すべきだが、検証機が一台のみであり、性能拡張を考慮するには他のサーバ筐体にまたがってデータベース・クラスタ化をすべきであり、適切な性能が得られない可能性があるため、機能上の確認のみとした。

1. 仮想環境から仮想サーバの作成（1CPU/2GB×2台）
  - \*システム基盤提供サービス(IaaS)を模した仮想環境を想定。
  - \*データベース・サーバ用のテンプレートを使用した。これは即時にデータベース・サーバを構築できるよう、あらかじめOS環境とデータベース環境が標準化された状態になっているもの（仮想サーバ・テンプレート）で、仮想サーバ作成時にこのテンプレートを指定することで、簡単に起動され、初期設定だけをするだけで利用ができるようになっている。
2. 仮想サーバの拡張
  - 仮想サーバを、受信用データベース・サーバにクラスタ環境として連結し、2台を追加することで受信用データベースを、3台構成に拡張する。
    - I. ネットワークへの仮想サーバの参加  
データベーステンプレートから作成されたデータベースをネットワークへ参加。
    - II. 管理コンソールから、追加データベース・サーバの確認  
ネットワークに参加した仮想サーバは、管理コンソール（Oracle Enterprise Manager）から確認を行う。
    - III. 管理コンソールから、データベース・クラスタへの追加

- 1台のみで稼働していたデータベースに、仮想サーバを追加。  
1台ずつ、合計2台を追加。
- IV. 管理コンソールから拡張確認  
管理コンソールから、サーバが3台になったことを確認。  
\*目視により、ビジュアルに判別することができる。
- 3. データベース動作確認  
3台にクラスタ化された受信データベースにアクセスを行い通常通りの動作ができるか確認を行う。
  - I. SQLインターフェースを用いて接続  
Oracle DatabaseのSQLインターフェース（SQLplus）を用いて、クラスタ化された受信データベースにアクセスを行う。  
接続時の、ログイン画面を目視し確認。
  - II. SQL文を用いて、受信データベースの表を確認  
SELECT文（表検索文）を用いて、受信データベースの表からデータベースから正しく出力されるか確認。（画面およびログによる出力確認。）

## C. 検証の結果

### a. 総評

バックアップ受信サーバは、仮想環境から作成されたデータベース・サーバを2台追加することにより3台構成のデータベース・クラスタとなり、停止することなく動的に性能を拡張できることが確認できた。

### b. 結果詳細

この検証では、直前までに行われた業務データ転送の検証に用いたデータベース（当初からの検証に用いたデータベース）を拡張対象として、仮想環境から作成された仮想サーバを段階的にクラスタリングすることにより、1サーバだったデータベース・サーバを3サーバ（3CPU/6GB）に拡張した。

なお当検証機のスペックでは、3台分のデータベース・クラスタとアプリケーションを立ちあげることのできるキャパシティがなく「京都府文書事務支援システム」の仮想環境を立ち上げることができず、データベース単体の作業となった。

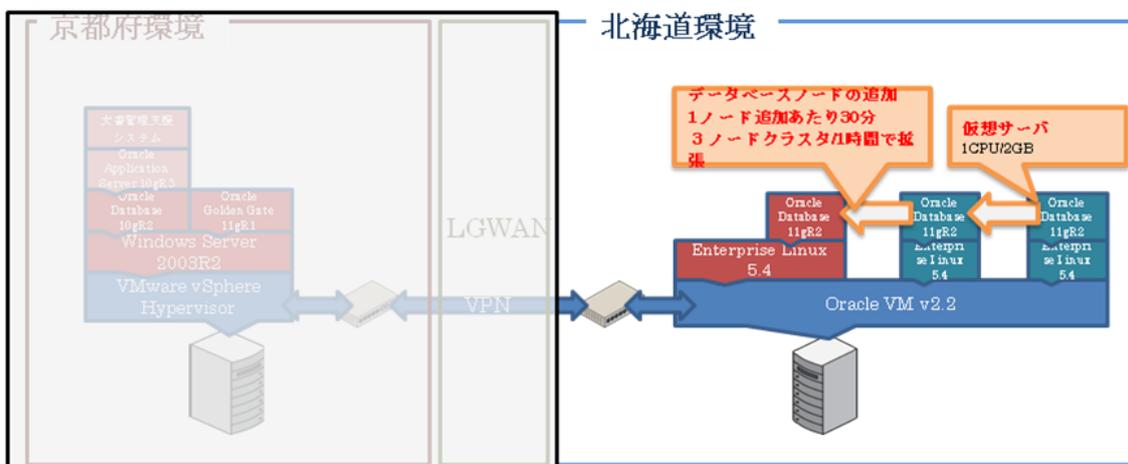


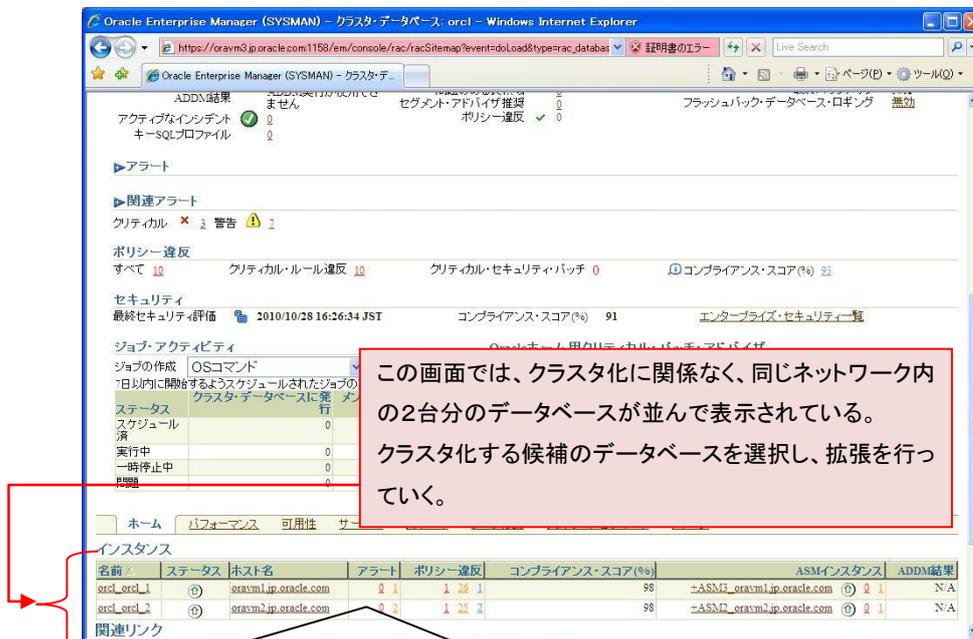
図 4-23 検証概要ならびに結果

一台あたり30分（サーバ作成で20分、クラスタ拡張作業で10分）で拡張することができた。今回の検証では2サーバ追加を行ない計1時間で完了することができた。1CPU/2GBのデータベース環境が拡張され、3CPU/6GBの環境に拡張することができた。

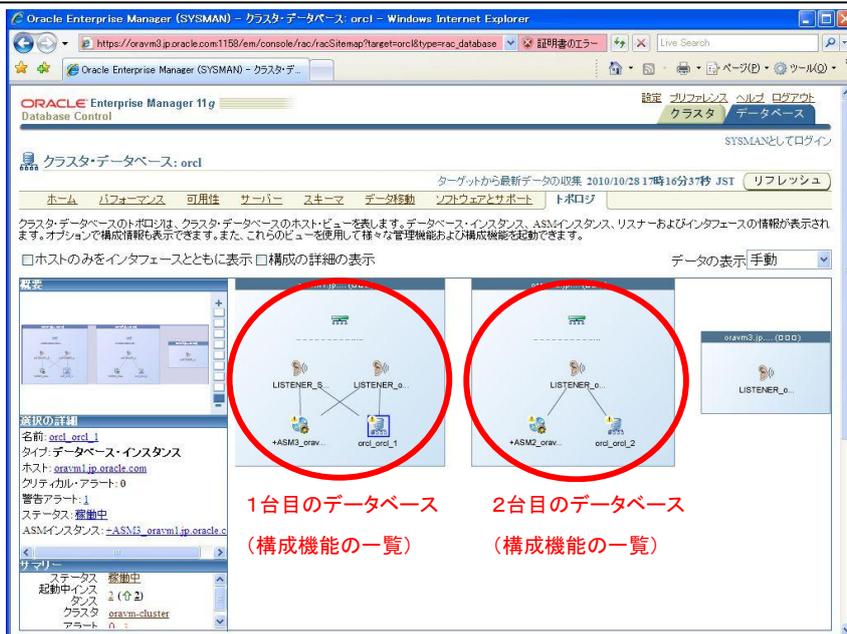
稼働の確認については、データベース管理コンソールから確認した他、行政機能データバックアップと同様、SQLインターフェース（SQLplusコマンド）を用いて接続し、実際に表が参照できるか確認を行った。クラスタ化されている状況であっても一台構成の場合と比べ設定変更することなく問題なく接続され、SELECT文による参照も問題は生じない結果となった。

ビジュアルな画面表示およびログの確認から無事3台になったことを確認でき、またSQLを通じたアクセスの結果通常のデータベースと同様に利用ができたことが確認できた。

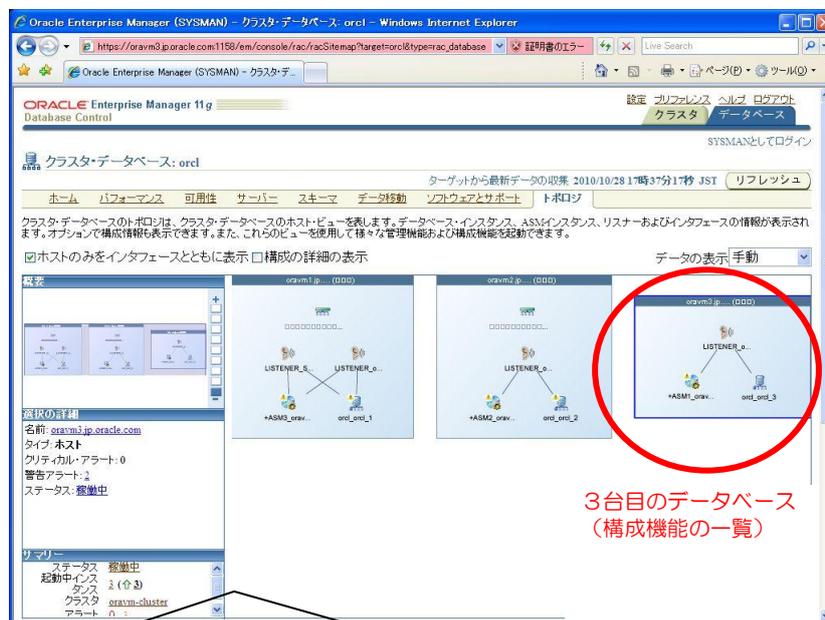
(作業内容・結果)



仮想環境で作成されたデータベース・サーバを確認している画面。  
クラスタ化の作業をおこなう最初の画面。



拡張中の状態表示(2サーバ)。当初1サーバしかなかったものが2台構成になったことをビジュアルに表現され、確実に動作が行えているか確認している画面。  
2サーバから、3サーバにするためにクラスタに属するサーバ数の最大値を2から3に変更。これにより、作成されたデータベース・サーバが動的にクラスタに参加され、3台での稼働を開始する。3台での並列実行により性能を向上するとともに、1サーバがダウンしても稼働し続ける環境が構築された。



最終的にデータベース・クラスタの拡張後、3サーバになっていることが確認できる。

#### D. 結果の考察

行政システム稼働環境（データベース・サーバ）の拡張性について、今回のデータベース・サーバ拡張検証では、1サーバあたり30分、2サーバを1時間で拡張させることが出来た。また、拡張作業中においてもデータベース・サーバは稼働しており、サービスの中断はなかった。この実証結果がもたらす、自治体クラウドにおける有用性と技術的な裏付けについて以下に考察を加える。

自治体クラウドの導入は自治体が持つ全てのシステムを一括して切り替えるビックバン方式ではなく、文書管理システムなど自治体間で共通性の高い業務から順次切り替えていく方式が有力である。また、自治体クラウドを利用する自治体も実証実験から始まる先行自治体から始まり、年度単位で徐々に増えていくことが予想される。このことから自治体クラウドにおいては、拡張性が重要なシステム要件の1つであると言える。

システムを構成するサーバのうち技術的に拡張が難しいのはアプリケーションサーバよりもデータベース・サーバであると一般的に言われている。これは、アプリケーションサーバ上にはプログラムを配置すればよいので1台毎が独立して動作できるのに対して、データベース・サーバ上にはデータが配置されるのでデータの同期や連携を考慮にいれた拡張方法が必要になるためである。

一般的にデータベース・サーバを拡張させる際には、拡張用のマシンをセットアップしデータの再配置を行わなければならないので多大な時間が必

要となる。また既存環境への追加作業を行う場合は、稼働しているデータベース・サーバを停止する必要がある。

本実証のように、クラウド環境から拡張ができた場合、ハードウェアならびにデータベースの調達から設置、拡張までの作業が即日にも行うことができる。物品調達が必要な場合、手配期間が長いこと、設置期間の長期を見込む必要があるが、これらを大幅に短縮できるほかより時間がかかり、スキルが必要なデータベース設定に時間をとる必要がなく大幅な時間短縮につながる。

表 4-21 データベース・サーバ拡張作業の手順比較（期間）

	一般的な拡張方法	実証手順
①ハードウェア見積	1ヶ月	1ヶ月
②ハードウェア手配	1ヶ月	即日
③ハードウェア設置	1週間	即日
④データベース設定	1ヶ月	即日
⑤データベース構築	1ヶ月	即日（動的拡張のため）
⑥新ハードウェアへの移行	1週間～2週間	即日（動的拡張のため）
（停止時間）	1日以上	無停止

本検証での実証のように、仮想化の技術を拡張性に求めた機能を利用することにより拡張を短時間に行うことができる。クラウド化するアプリケーションの追加や、利用自治体の追加が容易に行えることになる。このことは自治体クラウドの普及という面にも寄与することができる。また、サービス停止することなく拡張できる今回の方式は、住民向けサービスのクラウド化において特に有用と考える。

なお、本検証では、データベース・サーバの拡張機能として、Oracle Databaseの備える“サーバ・プール”機能を用いている。“サーバ・プール”機能ではデータベース・システムで利用されるサーバ資源（物理的なサーバ、CPU、メモリもしくは仮想環境）を動的に管理し、今回のような拡張の際には適切に負荷分散、可用性をサーバ・プールにあるサーバ資源を積極的に活用することで実現できるものである。

#### ウ) 災害対策サイト（iDC）における行政機能（業務サービス）

復旧バックアップサイトへアプリケーション切り替え（障害発生時）

##### A. 実証の概要・目的

ここでは、業務データの逐次のバックアップを行う通常運用時のオフサイトバックアップの運用イメージに加え、障害発生時の、バックアップサイト上での業務アプリケーション切り替えを中心とした業務復旧に焦点を定め、円滑な業務復旧を行う実証を行う。

本番サイト（京都府）が利用不可能な状況に陥った場合を想定し、バックアップサイト（北海道）において本番サイトと同じ業務アプリケーションを

起動し、既に通常運用のデータ転送にてバックアップしていたデータを利用することにより、行政業務機能の即時復旧を目的とした検証を行う。

ここでは「京都府文書事務支援システム」のアプリケーションを利用し、より実用性・実現性の高い行政機能復旧検証を目指す。

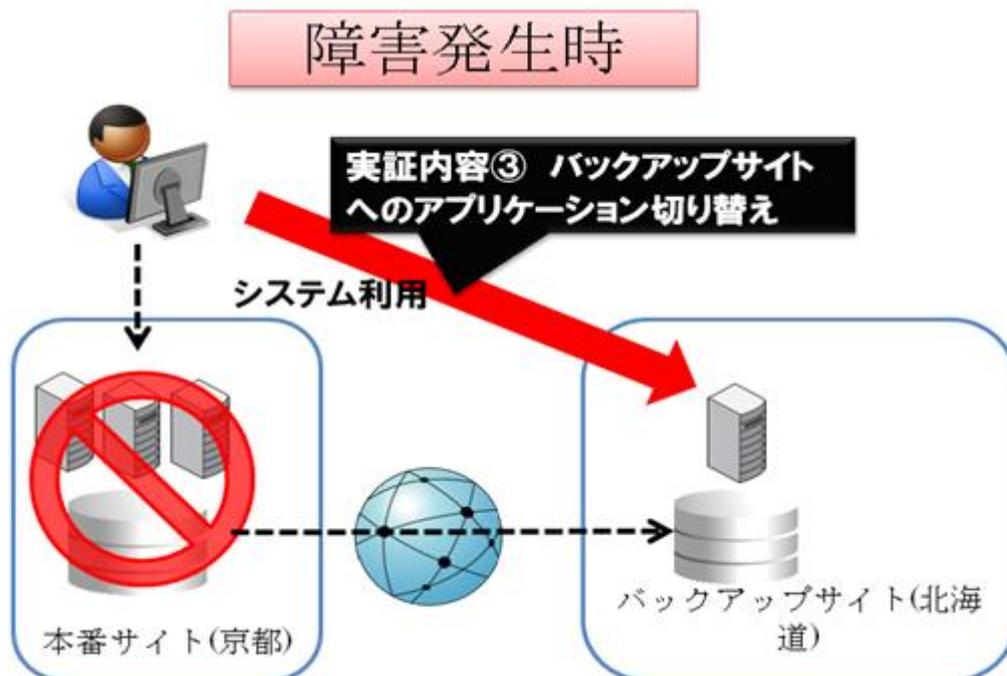


図 4-24 バックアップサイトへのアプリケーション切り替え

## B. 実証の内容

京都府・北海道間のバックアップ・データベース環境へ、それぞれ仮想化環境から起動された“京都府文書事務支援システム”を接続し、動作確認を行った。

文書事務支援システムのデータベースで、実際に稼働しているデータを保管しているテーブル（2テーブル）をすべて京都・北海道でリンクし、実証を行った。京都府環境では、アプリケーションからデータの登録を行い、北海道側では仮想環境から立ち上げた同じく文書事務支援システムを、受信サーバに接続を行うことで確認している。

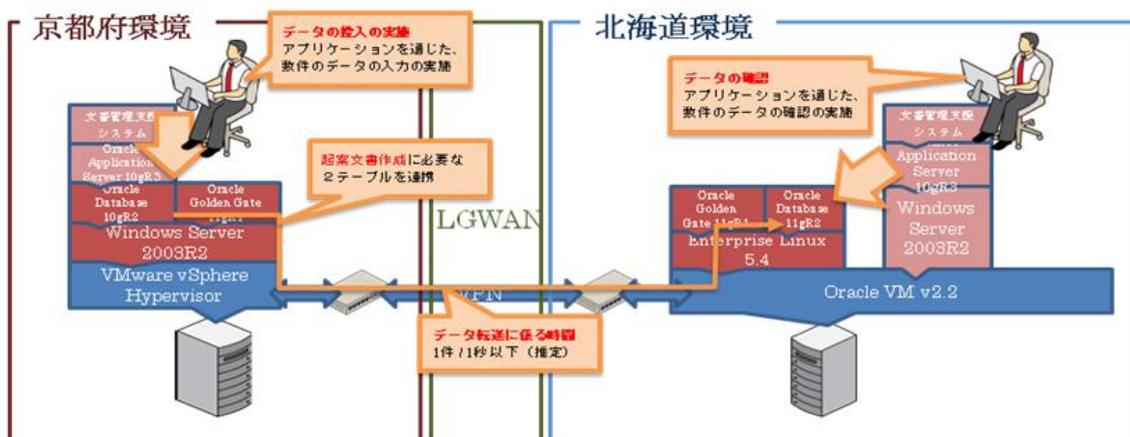


図 4-25 復旧バックアップサイトへアプリケーション切り替え実証イメージ

業務アプリケーションからの確認

1. 実証1で用意の表のほか、関連表計2表を用意。(合計およそ20MB)  
\*「起案文書作成」に必要な表群
2. 京都環境/北海道環境にそれぞれ、同表を配置。
3. 京都環境/北海道環境の各データベースの各表同士の連結を行う。  
(転送機能Golden Gateによる設定、実証1設定に基づく)
4. 京都府側の「文書事務支援システム」の「起案文書作成」メニューから手動により操作を行い、順次データを入力。  
(およそ100byte~200byte/1件)
5. リアルタイム同期設定。  
京都府のデータベース更新を逐次監視し、更新ごと転送を行うように設定。
6. 北海道側の「文書事務支援システム」でデータの確認を行う。  
北海道環境の京都府文書事務支援システムのユーザインターフェースからの確認作業。

C. 実証の結果

a. 総評

本番サイト（京都府）で操作した文書事務支援システムの実行中の業務データが逐次、バックアップサイト（北海道）にバックアップされることが確認できた。続いて、バックアップサイト（北海道）環境において、バックアップされたデータが即時アプリケーションから参照することができた。

なおデータは更新ごとリアルタイムに北海道に送られデータに差は生じていない。

b. 結果詳細

行政機能データバックアップ検証結果から、VPN回線に問題のない少量のデータであればデータ転送について問題はなく、少量のデータのみ流れる業務であれば差し障りなく実証が可能との判断から、北海道環境で実際のデータを用いた確認が可能であると考え、実証した。

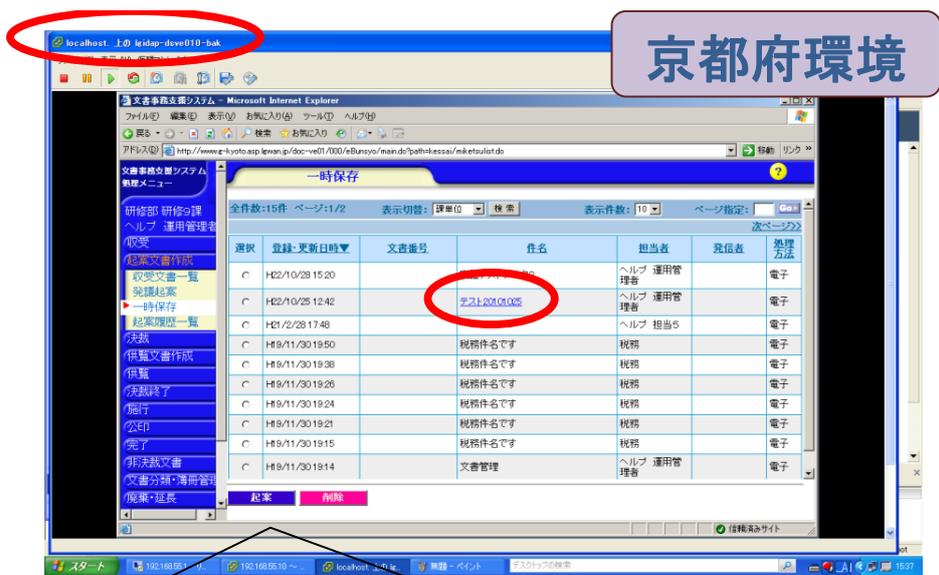
実証内容としては、「京都府文書事務支援システム」のオペレーションを中心に、京都府で操作した内容が正しく北海道環境の同環境にて同じデータが確認できるか、データが正しく転送されていることを確認することとし、リアルタイムに確認が可能か検証を行っている。

検証にもちいた機能は「起案文書作成」であり、これに用いられている2テーブル（約20MB）を京都府、北海道のデータベースに連携させることにより実証を行った。

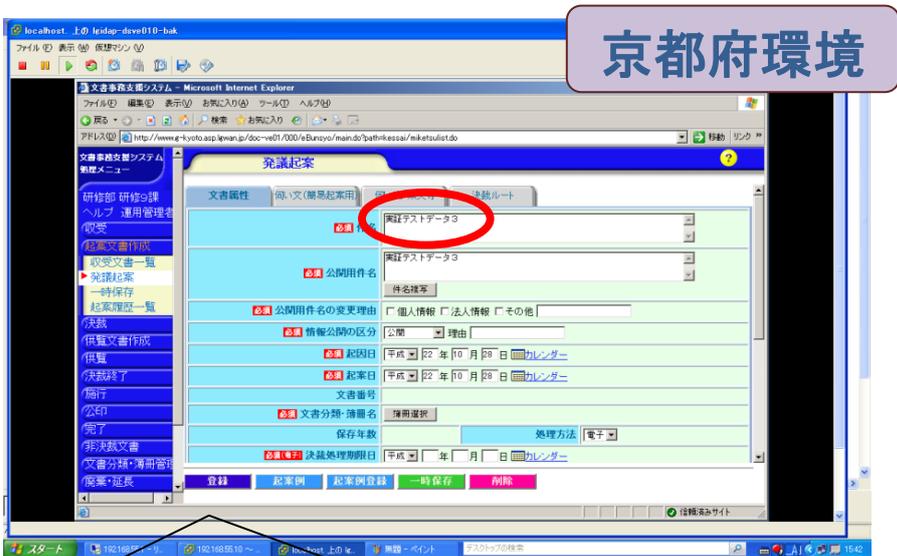
検証に利用した、行政機能環境については、京都府環境ではバックアップに用いた京都府文書事務支援システムの環境を使用、北海道環境ではバックアップ受信サーバのデータベースに対し、新たに仮想環境から起動された京都府文書事務支援システムをバックアップ・データベースに接続し、データを参照するよう設定することで本検証を実施できるようにしている。

なお、北海道環境での文書事務支援システム起動は仮想環境から行ったものだが、OS環境を起動させ、次いで文書事務支援システムを起動、利用可能になるまで10分程度で立ち上がり、即時に利用可能となった。

京都府、北海道双方の表の連結後の検証としまして京都府環境より実際の起案文書作成を行うことでデータを投入している。京都府の検証では、仮想環境に用いているVMware vSphere Hypervisorで利用できるリモートコンソール機能を用いて、サーバ上のWebブラウザを開くことで「京都府文書事務支援システム」へ接続、作業を行っている。

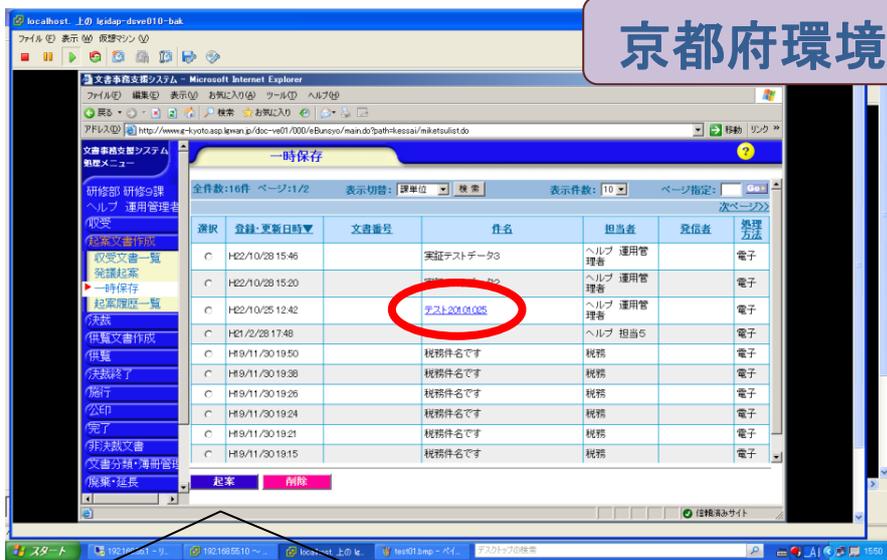


VMware 管理コンソールから、サーバ上の Internet Explorer を開きシステムにアクセスすることで検証を行っている画面。  
この画面では検証に用いた起案文書作成画面をひらいている。ここでは最初のテストデータとして「テストデータ2」が入力されていることを確認している。



起案文書作成から、テストデータ2に続くデータとして、テストデータ3を入力している。

京都府環境



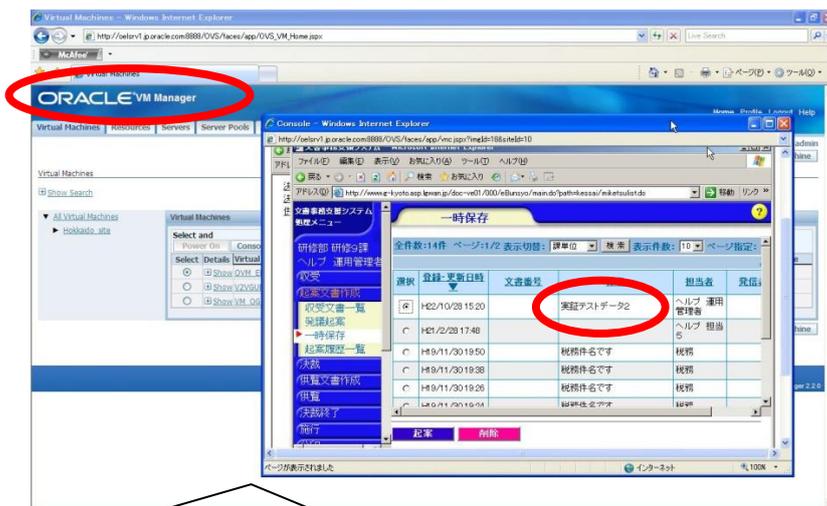
テストデータ3が登録されていることが確認できた。

ここまでの京都府の操作によりテストデータが転送されていると、想定している。

続いて、北海道側の作業として京都府で作業したデータ2, 3がそれぞれ段階的に反映されているか確認する。

ここでは、仮想環境であるOracle VMの管理コンソールを用いて京都府と同様にWebブラウザを用いて、「京都府文書事務支援システム」にアクセスすることでデータの確認を行っている。

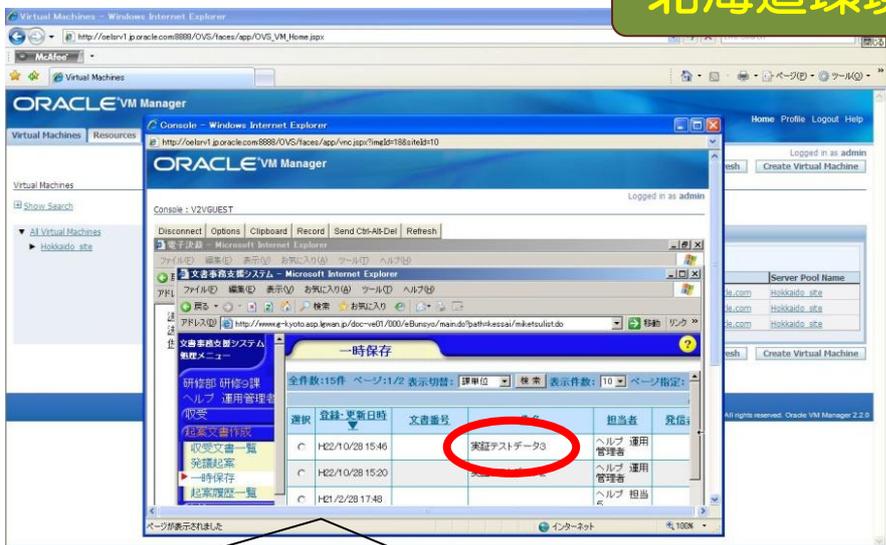
北海道環境



動作確認に用いた、OracleVM の管理コンソール画面。ここから操作画面を開き、Internet Explorerを立ち上げることで確認を行う。開いた時点で、京都府操作で入力されたテストデータ2が参照できている。

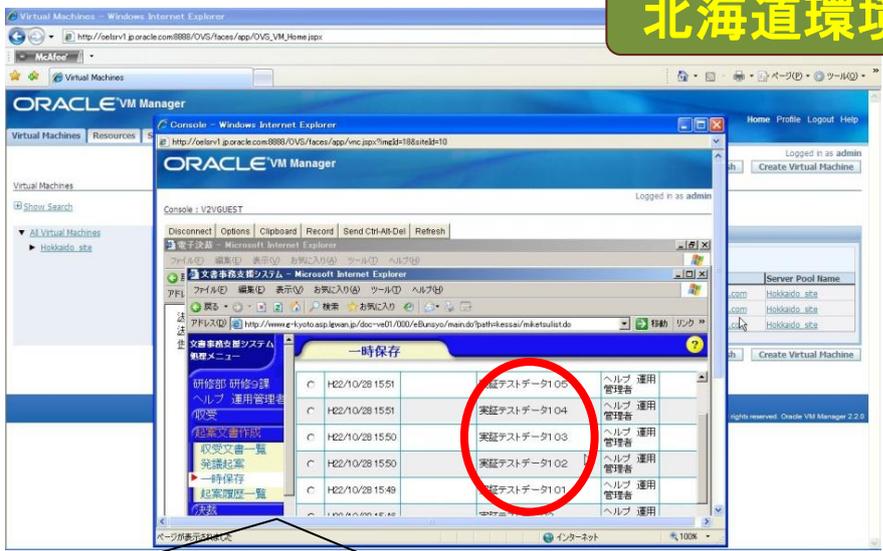
続いて、京都府側の追加のデータ操作に合わせてデータが確認できるか、確認作業を行う。

北海道環境



京都府でのテストデータ3の入力後、即時に反映されたことを北海道でも確認。

北海道環境



以降、京都側から登録されたデータを確認。(101~105)

いずれも京都作業後、即時北海道での更新が反映され確認できた。

## D. 結果の考察

### a. データの転送状況について

検証1の結果から、データ転送に当たっては本検証のようなデータサイズ（100byte - 200byte）であっても1秒～10秒程度のタイムラグが生じており、転送中にネットワークの寸断やサーバ停止した場合、停止時間に応じて差分が生じるおそれがある。

寸断が生じた場合停止時点から転送されたデータの番号を記録し、復旧後に再転送する機能を有効にすることでバックアップサイトへの伝搬漏れをおこさせない仕組みが必要と考える。なお実証に使用した転送機能（Golden Gate）ではこの機能があり対処可能であった。

### b. 業務アプリケーションの復旧について

本番サイト（京都府）で入力した文書事務支援システムのデータが逐次バックアップサイト（北海道）に転送され、バックアップサイト側で起動したアプリケーションから参照できたという検証結果を2つの観点で考察する。

1つは、本番サイトにおいて業務アプリケーションから入力された業務データが、「逐次」、バックアップサイトに転送されたという点について、これは本番サイトが地震などの大規模災害に見舞われた場合に、災害の直前まで処理されていた業務データが遠隔地でバックアップできていることを意味する。

災害対策サイトを評価する指針の1つに目標復旧時点（RPO：recovery point objective）があるが、これは災害の発生によるシステム停止時に、どの時点までさかのぼってデータを回復させるかを表している。今回の実証実験の結果で言えば、復旧時点は「被災直前のデータ」と言うことができる。これは、極めて高いデータの信頼性が求められる自治体クラウドにおいても十分な検証結果であると評価できる。

もう1つの観点は、バックアップサイト側での業務切り替えに要した時間であり、データベース・サーバの拡張と同様に、本検証においても仮想化技術を用いてアプリケーションの環境構築と起動を行った。これにより、あらかじめセットアップ済みの仮想化アプリケーションをバックアップサイトに即時に展開させることができた。

災害対策サイトを評価する指標である目標復旧時間（RTO：recovery time objective）は、災害や障害による業務／情報システムの停止から、定められたレベルにサービスが復旧するまでに必要となる経過時間を表し、今回の実証実験の結果では、OSが起動し、ついでアプリケーションが起動、利用できるまで10分で完了しており、復旧時間は「10分以内」と言うことができる。

以上の事から、今回の実証実験で採用したデータ転送方式、及びアプリ

ケーション仮想化技術によって、本番サイトが被災してから10分以内にバックアップサイト側で業務切り替えを行えることが確認できた。

### c. 実証全体に関する考察

実証では災害が発生した際、即時に行政機能を復旧させることを観点に検証を行っている。

クラウド化された自治体共同利用にはハードウェア・システムの共用化による調達コストの削減効果、運用の共通化によるコスト負担の軽減が期待されている。しかし仮に地域ごと一拠点ずつ共同利用型でシステムの取りまとめを行ったとした場合、複数自治体が行政システムを稼働している状況で災害が発生した場合、多数の行政システムに被害が及ぶ可能性が高い。

まず行政機能を早期に回復させるためにシステムに求められることは、災害発生前までに行われたデータ更新が確実にバックアップされていることがまず挙げられる。このデータは即時利用できなければならない。データが確実になければたとえアプリケーションが復旧できたとしても行政機能を復旧することにはならない。また共同利用されるシステム群については自治体ごと、目的とする行政業務に合わせて調達がなされることで様々な種類のシステム構成（OSの違い、データベースの違い、実行環境の違い等）があり、これらを確実にバックアップする必要がある。

実証では、確実なバックアップができたことを確認できた上で、バックアップデータを扱うようアプリケーションの起動を行い無事に起動され、直前までの状態をもって利用することができた。

システムの形態としては一般的な三層アプリケーション（Web型アプリケーション）形態を用いており同様の構成であれば実証のような行政機能バックアップが可能であることが証明されたと考える。また行政システム環境については、異なるOS環境（WindowsからLinux）、異なるバージョン環境（Oracle 10gR2からOracle 11gR2）を用意しバックアップを実現できた。このことは複数自治体システムがあったとして、バックアップを実現する手段になりえることを証明されたと考えられる。

実証環境構築については自治体共同利用を観点にクラウドの環境を模している。

一般にクラウドの利点として、コスト効果・俊敏性（短期間でシステム構築ができること）が挙げられている。実証における構築面でのクラウド効果としては1台のみの検証機環境からバックアップ・データベースおよびアプリケーション環境を普段稼働していないところから起動することができた。このことは常時稼働の待機系システムを用意する必要がないことを示している結果であり、仮に複数の行政アプリケーションがクラウド化されたと考えた場合、待機系システムが大幅に削減されることになることと結論づけることができる。

実証から、クラウド環境で構築されたバックアップサイトは最低台数でのバックアップ稼働が可能であり、システム復旧もクラウド環境から即座に行えることが証明された。自治体共同利用にクラウドを実装することは、特にバックアップサイトでは即座にメリットが生じ有効性が高いことが証明されたと考えられる。

(2) オンサイトバックアップとオフサイトバックアップの比較

ア) オンサイトバックアップ

A. 実証の概要・目的

【概要】

データ送受信サーバの仮想化環境上に構築した電子申請システムのデータベースバックアップ・リストアを実施し、各作業の処理時間及びテープバックアップ速度を取得する。実証イメージを図 4-26 に示す。

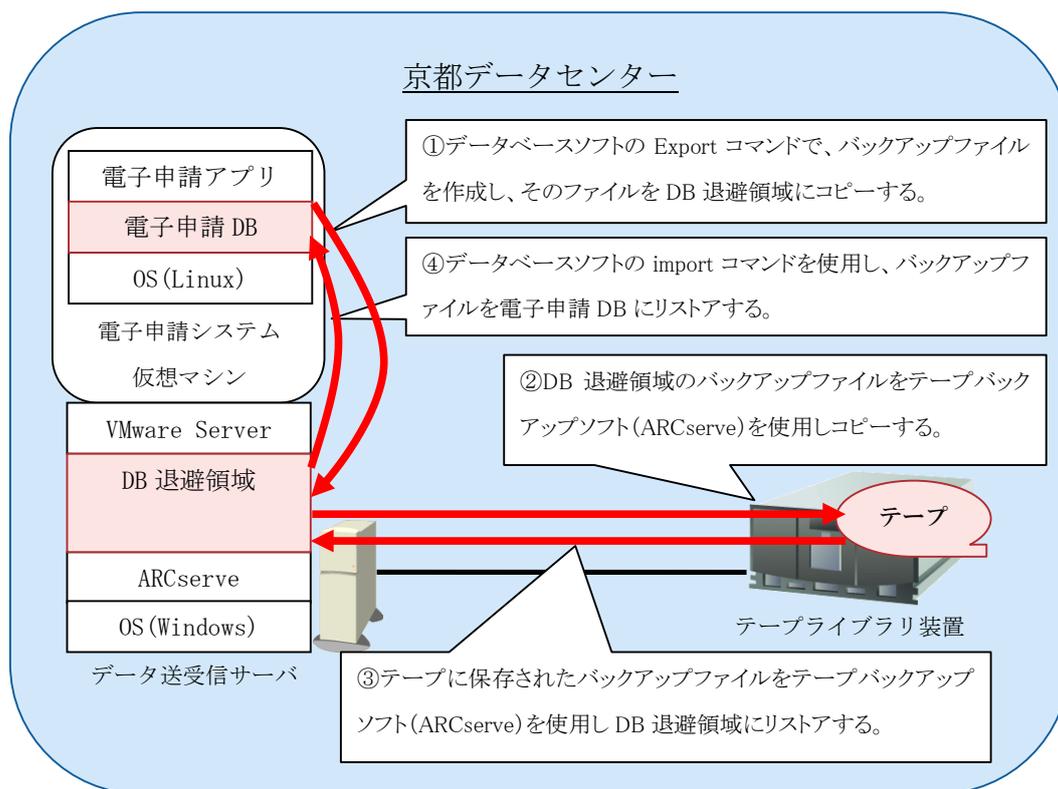


図 4-26 オンサイトバックアップ実証イメージ

【目的】

電子申請システムのデータベースバックアップ・リストアに要した時間を計測し、オフサイトバックアップ・リストアの有効性・効率性を確認するための基礎値として利用する。

B. 実証の内容

【前提条件】

a. オンサイトバックアップ対象データベースについて

オンサイトバックアップ対象データベースは、「自治体コンピューティング」実証実験でデータ送受信サーバの仮想化環境に構築した電子申請シ

システムのデータベースを対象とする。電子申請システムのデータベースは「SymfoWARE Server」で構築されており表 4-22 に示す環境で構成されている。また、オフサイトバックアップ時には表 4-23 に示すデータが登録されている。

表 4-22 電子申請データベース環境

項目	内容
OS	Red Hat Enterprise Linux ES(v.3 for x86)
データベースソフト	Symfoware Server Enterprise Edition V6.0L10
データベースサイズ	22GB
テーブル数	49 (マスタ系テーブル:25 トランザクション系テーブル:24)

表 4-23 オフサイトバックアップ時データ登録状況

テーブル	データ登録数
マスタ系テーブル 1	381
マスタ系テーブル 2	714
マスタ系テーブル 3	46,465
マスタ系テーブル 4	19,493
マスタ系テーブル 5	159
マスタ系テーブル 6	25,519
マスタ系テーブル 7	2
マスタ系テーブル 8	2,870
マスタ系テーブル 9	35
マスタ系テーブル 10	392
マスタ系テーブル 11	121,684
マスタ系テーブル 12	21,331
マスタ系テーブル 13	9
マスタ系テーブル 14	61
マスタ系テーブル 15	268
マスタ系テーブル 16	1,438
マスタ系テーブル 17	1,434
マスタ系テーブル 18	11,035
マスタ系テーブル 19	8,207
マスタ系テーブル 20	610
マスタ系テーブル 21	30,567
マスタ系テーブル 22	61
マスタ系テーブル 23	862
マスタ系テーブル 24	648
マスタ系テーブル 25	243
トランザクション系テーブル 1	2,107
トランザクション系テーブル 2	5,320
トランザクション系テーブル 3	2,209

テーブル	データ登録数
トランザクション系テーブル 4	1
トランザクション系テーブル 5	9,930
トランザクション系テーブル 6	12,037
トランザクション系テーブル 7	2,544
トランザクション系テーブル 8	17,681
トランザクション系テーブル 9	64
トランザクション系テーブル 10	128
トランザクション系テーブル 11	162
トランザクション系テーブル 12	87
トランザクション系テーブル 13	13
トランザクション系テーブル 14	11
トランザクション系テーブル 15	34,544
トランザクション系テーブル 16	12
トランザクション系テーブル 17	49
トランザクション系テーブル 18	2,709
トランザクション系テーブル 19	466
トランザクション系テーブル 20	1
トランザクション系テーブル 21	3
トランザクション系テーブル 22	2
トランザクション系テーブル 23	1
トランザクション系テーブル 24	1

b. データベースバックアップ・リストア方法について

電子申請データベースのバックアップデータの作成は「SymfoWARE Server」のexportコマンド(rdbunl)で実施し、リストアはimportコマンド(rdbsloader)で実施する。データベースバックアップ・リストアイメージを図 4-27 に示す。

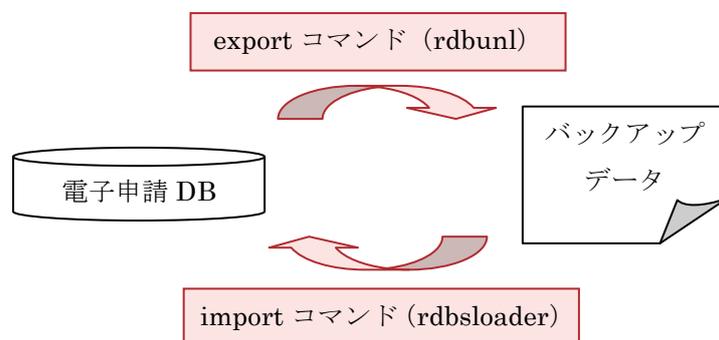


図 4-27 データベースバックアップリストアイメージ

rdbunl / rdbsloader コマンドはテーブルに対して、高速にデータを追加及び吸い上げる機能を備えており、外部ファイルはバイナリ形式又はCSV形式に対応しているという特長を持っている。本実証

実験では、バックアップ対象となる電子申請データベースにXMLファイルが格納されているためバイナリ形式で作業を実施する。

### c. テープライブラリ装置バックアップ・リストア方法について

テープライブラリ装置バックアップ・リストアはデータ送受信サーバに導入されているバックアップソフト「CA ARCserve Backup」で実施する。

#### 【実施手順】

前提条件を踏まえ、本実証実験では表 4-24 に示す作業手順でオンサイトバックアップ・リストアを実施し、各作業で出力されたデータのファイルサイズ及び各作業の処理時間、テープバックアップ速度を取得する。

表 4-24 オンサイトバックアップ・リストア実施手順

手順	作業概要	作業内容
1	バックアップデータ作成	データベースソフト(SymfoWARE)の export コマンドで、電子申請システム仮想マシンの電子申請データベースバックアップデータを作成し、データ送受信サーバのDB退避領域にコピーする。
2	バックアップデータのテープバックアップ	データ送受信サーバのDB退避領域に存在する電子申請データベースのバックアップデータをテープバックアップソフト (ARCserve) で、テープライブラリ装置のテープにバックアップする。
3	バックアップデータのテープバックリストア	テープライブラリ装置のテープにバックアップした電子申請データベースのバックアップデータをテープバックアップソフト (ARCserve) で、データ送受信サーバのDB退避領域にリストアする。
4	バックアップデータのデータベースリストア	データ送受信サーバのDB退避領域に存在する電子申請データベースのバックアップデータをデータベースソフト(SymfoWARE)の import コマンドで、電子申請データベースにリストアする。

### C. 実証の結果

今回実施したオンサイトバックアップ・リストアの結果及び各作業の所要時間を表 4-25 に示す。

表 4-25 オンサイトバックアップ・リストアの所要時間

手順	作業概要	作業内容	結果	所要時間
1	バックアップデータ作成	データベースソフト(SymfoWARE)の export コマンドで、電子申請データベースのバックアップデータを作成し、データ送受信サーバのDB退避領域にコピーする。	OK	6分56秒
2	バックアップデータのテープバックアップ	データ送受信サーバのDB退避領域に存在する電子申請データベースのバックアップデータをテープバックアップソフト (ARCserve) で、テープライブラリ装置のテープにバックアップする。	OK	7分30秒
3	バックアップデータのテープバックリストア	テープライブラリ装置のテープにバックアップした電子申請データベースのバックアップデータをテープバックアップソフト (ARCserve)	OK	2分56秒

手順	作業概要	作業内容	結果	所要時間
		e) で、データ送受信サーバのDB退避領域にリストアする。		
4	バックアップデータのデータベースリストア	データ送受信サーバのDB退避領域に存在する電子申請データベースのバックアップデータをデータベースソフト(SymfoWARE)の import コマンドで、電子申請データベースにリストアする。	OK	29 分 39 秒
合計				47 分 01 秒

オンサイトバックアップ・リストア作業手順1～4実施時に出入力されたバックアップデータのファイルサイズを表 4-26 に示す。

表 4-26 バックアップファイルサイズ

テーブル	手順 1	手順 2	手順 3	手順 4
	ファイルサイズ	ファイルサイズ	ファイルサイズ	ファイルサイズ
マスタ系テーブル 1	84KB	84KB	84KB	84KB
マスタ系テーブル 2	212KB	212KB	212KB	212KB
マスタ系テーブル 3	5,630KB	5,630KB	5,630KB	5,630KB
マスタ系テーブル 4	2,739KB	2,739KB	2,739KB	2,739KB
マスタ系テーブル 5	12KB	12KB	12KB	12KB
マスタ系テーブル 6	2,536KB	2,536KB	2,536KB	2,536KB
マスタ系テーブル 7	2KB	2KB	2KB	2KB
マスタ系テーブル 8	316KB	316KB	316KB	316KB
マスタ系テーブル 9	4KB	4KB	4KB	4KB
マスタ系テーブル 10	33,746KB	33,746KB	33,746KB	33,746KB
マスタ系テーブル 11	19,179KB	19,179KB	19,179KB	19,179KB
マスタ系テーブル 12	4,302KB	4,302KB	4,302KB	4,302KB
マスタ系テーブル 13	3KB	3KB	3KB	3KB
マスタ系テーブル 14	10KB	10KB	10KB	10KB
マスタ系テーブル 15	60KB	60KB	60KB	60KB
マスタ系テーブル 16	208KB	208KB	208KB	208KB
マスタ系テーブル 17	256KB	256KB	256KB	256KB
マスタ系テーブル 18	4,608KB	4,608KB	4,608KB	4,608KB
マスタ系テーブル 19	1,066KB	1,066KB	1,066KB	1,066KB
マスタ系テーブル 20	55KB	55KB	55KB	55KB
マスタ系テーブル 21	5,344KB	5,344KB	5,344KB	5,344KB
マスタ系テーブル 22	6KB	6KB	6KB	6KB
マスタ系テーブル 23	517KB	517KB	517KB	517KB
マスタ系テーブル 24	101KB	101KB	101KB	101KB
マスタ系テーブル 25	10KB	10KB	10KB	10KB
トランザクション系テーブル 1	1,200KB	1,200KB	1,200KB	1,200KB
トランザクション系テーブル 2	528,659KB	528,659KB	528,659KB	528,659KB
トランザクション系テーブル 3	358,363KB	358,363KB	358,363KB	358,363KB

テーブル	手順 1	手順 2	手順 3	手順 4
	ファイルサイズ	ファイルサイズ	ファイルサイズ	ファイルサイズ
トランザクション系テーブル 4	2KB	2KB	2KB	2KB
トランザクション系テーブル 5	3,373KB	3,373KB	3,373KB	3,373KB
トランザクション系テーブル 6	1,968KB	1,968KB	1,968KB	1,968KB
トランザクション系テーブル 7	2,432KB	2,432KB	2,432KB	2,432KB
トランザクション系テーブル 8	2,686KB	2,686KB	2,686KB	2,686KB
トランザクション系テーブル 9	21KB	21KB	21KB	21KB
トランザクション系テーブル 10	561KB	561KB	561KB	561KB
トランザクション系テーブル 11	18KB	18KB	18KB	18KB
トランザクション系テーブル 12	8KB	8KB	8KB	8KB
トランザクション系テーブル 13	2KB	2KB	2KB	2KB
トランザクション系テーブル 14	2KB	2KB	2KB	2KB
トランザクション系テーブル 15	20,310KB	20,310KB	20,310KB	20,310KB
トランザクション系テーブル 16	1,134KB	1,134KB	1,134KB	1,134KB
トランザクション系テーブル 17	9,062KB	9,062KB	9,062KB	9,062KB
トランザクション系テーブル 18	2,062KB	2,062KB	2,062KB	2,062KB
トランザクション系テーブル 19	303,065KB	303,065KB	303,065KB	303,065KB
トランザクション系テーブル 20	1KB	1KB	1KB	1KB
トランザクション系テーブル 21	1KB	1KB	1KB	1KB
トランザクション系テーブル 22	2KB	2KB	2KB	2KB
トランザクション系テーブル 23	2KB	2KB	2KB	2KB
トランザクション系テーブル 24	2KB	2KB	2KB	2KB
合計	1,315,942KB	1,315,942KB	1,315,942KB	1,315,942KB

オンサイトバックアップ・リストア作業前及び手順 4 実施後のデータ登録数を表 4-27 に示す。

表 4-27 データベースデータ登録数

テーブル	作業前	手順 4 実施後
	データ登録数	データ登録数
マスタ系テーブル 1	381	381
マスタ系テーブル 2	714	714
マスタ系テーブル 3	46,465	46,465
マスタ系テーブル 4	19,493	19,493
マスタ系テーブル 5	159	159
マスタ系テーブル 6	25,519	25,519
マスタ系テーブル 7	2	2
マスタ系テーブル 8	2,870	2,870
マスタ系テーブル 9	35	35
マスタ系テーブル 10	392	392
マスタ系テーブル 11	121,684	121,684
マスタ系テーブル 12	21,331	21,331

テーブル	作業前	手順 4 実施後
	データ登録数	データ登録数
マスタ系テーブル 13	9	9
マスタ系テーブル 14	61	61
マスタ系テーブル 15	268	268
マスタ系テーブル 16	1,438	1,438
マスタ系テーブル 17	1,434	1,434
マスタ系テーブル 18	11,035	11,035
マスタ系テーブル 19	8,207	8,207
マスタ系テーブル 20	610	610
マスタ系テーブル 21	30,567	30,567
マスタ系テーブル 22	61	61
マスタ系テーブル 23	862	862
マスタ系テーブル 24	648	648
マスタ系テーブル 25	243	243
トランザクション系テーブル 1	2,107	2,107
トランザクション系テーブル 2	5,320	5,320
トランザクション系テーブル 3	2,209	2,209
トランザクション系テーブル 4	1	1
トランザクション系テーブル 5	9,930	9,930
トランザクション系テーブル 6	12,037	12,037
トランザクション系テーブル 7	2,544	2,544
トランザクション系テーブル 8	17,681	17,681
トランザクション系テーブル 9	64	64
トランザクション系テーブル 10	128	128
トランザクション系テーブル 11	162	162
トランザクション系テーブル 12	87	87
トランザクション系テーブル 13	13	13
トランザクション系テーブル 14	11	11
トランザクション系テーブル 15	34,544	34,544
トランザクション系テーブル 16	12	12
トランザクション系テーブル 17	49	49
トランザクション系テーブル 18	2,709	2,709
トランザクション系テーブル 19	466	466
トランザクション系テーブル 20	1	1
トランザクション系テーブル 21	3	3
トランザクション系テーブル 22	2	2
トランザクション系テーブル 23	1	1
トランザクション系テーブル 24	1	1

オンサイトバックアップ・リストア手順 2 実施時のテーブルバックアップスルーット及びオンサイトバックアップ・リストア手順 3 実施時のテーブルリストアスルーットを表 4-28 に示す。このスルーット値はバックアップ

ソフト「CA ARCserve Backup」が算出した値である。

表 4-28 テープバックアップリストア平均スループット

処理	平均スループット
テープバックアップ	3,876.09 MB/分
テープリストア	4,034.50 MB/分

#### D. 結果の考察

データベースのバックアップ・リストアは47分01秒で実施でき、バックアップ手順1～4実施時のバックアップデータファイルサイズ及びバックアップ前と手順4のリストア後のデータ登録数が同数であったことから、データベースのバックアップ・リストアは正常終了したことが言える。オンサイトバックアップ・リストアに要した時間については、手順4の電子申請データベースのリストアに多少時間がかかった。これは電子申請データベースをVMware上に作成しているためにディスク負荷がボトルネックになっているためだと考えられる。データベースを仮想マシン上に作成する場合は、高性能ハードディスクの選定やデータベースの配置ディスクの分散等を考慮すべきだと思われる。

今回の実証実験で、バックアップは約15分、リカバリは約30分で完了したので、オンサイトバックアップ・リストアは有効な手段だと考えられる。

#### イ) オフサイトバックアップ

##### A. 実証の概要・目的

###### 【概要】

##### a. オフサイトバックアップ・リストア

京都データセンターの電子申請データベースをLGWAN経由（VPNによるトンネリング通信）で北海道データセンターのバックアップサーバにオフサイトバックアップ・リストアする。また、各作業の処理時間及びファイル転送速度を計測する。

オフサイトバックアップ・リストア実証イメージを図 4-28 に示す。

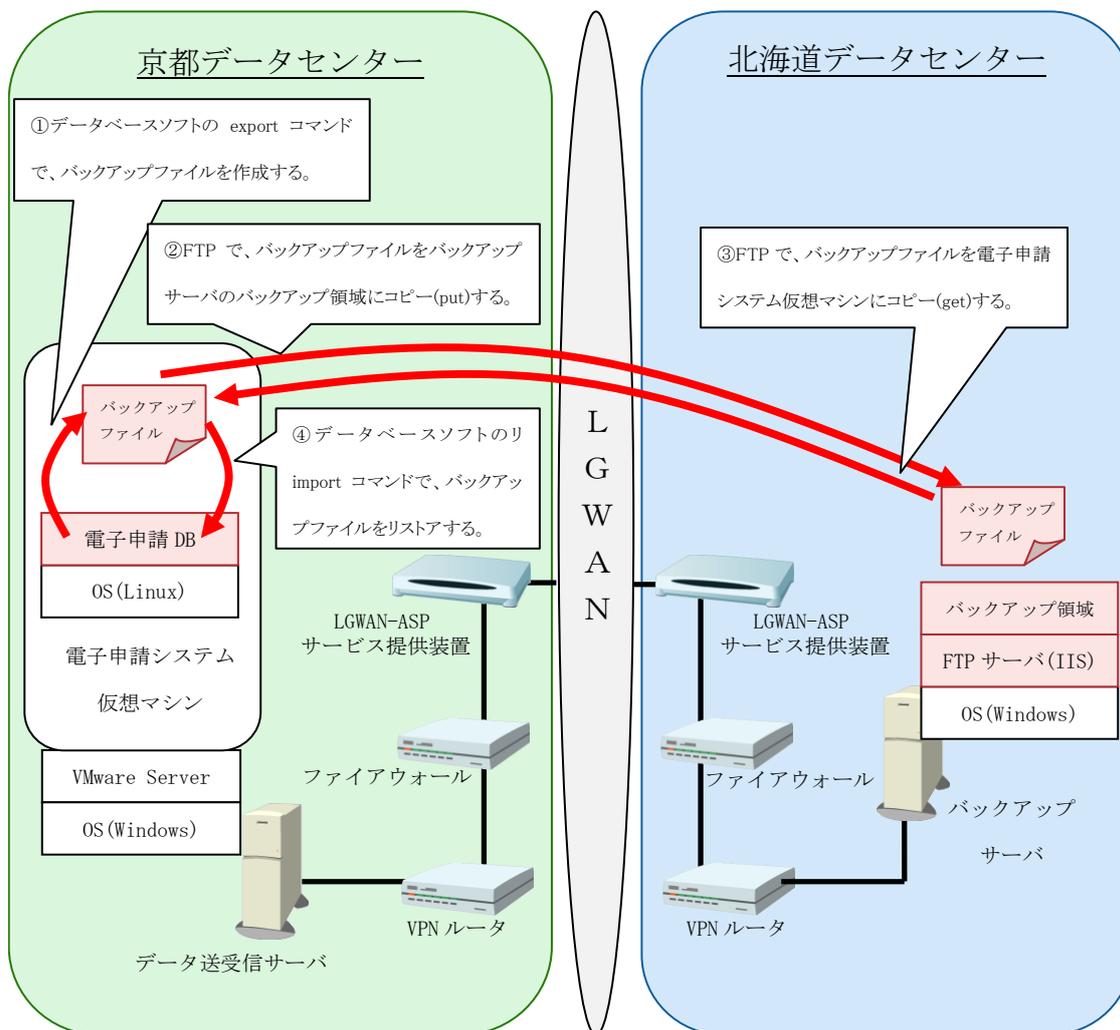


図 4-28 オフサイトバックアップ・リストア実証イメージ

b. レプリケーション

電子申請システム仮想マシンに対して申請処理（30秒に1件）を実施する。次にデータベースの差分データを一定間隔（12時間、2時間、5分の3パターン）で北海道データセンターのバックアップサーバに転送し、差分データをデータベースに反映する。また、各作業の処理時間、ファイル転送速度及び各サーバの性能情報を計測する。

レプリケーション実証イメージを図 4-29 に示す。

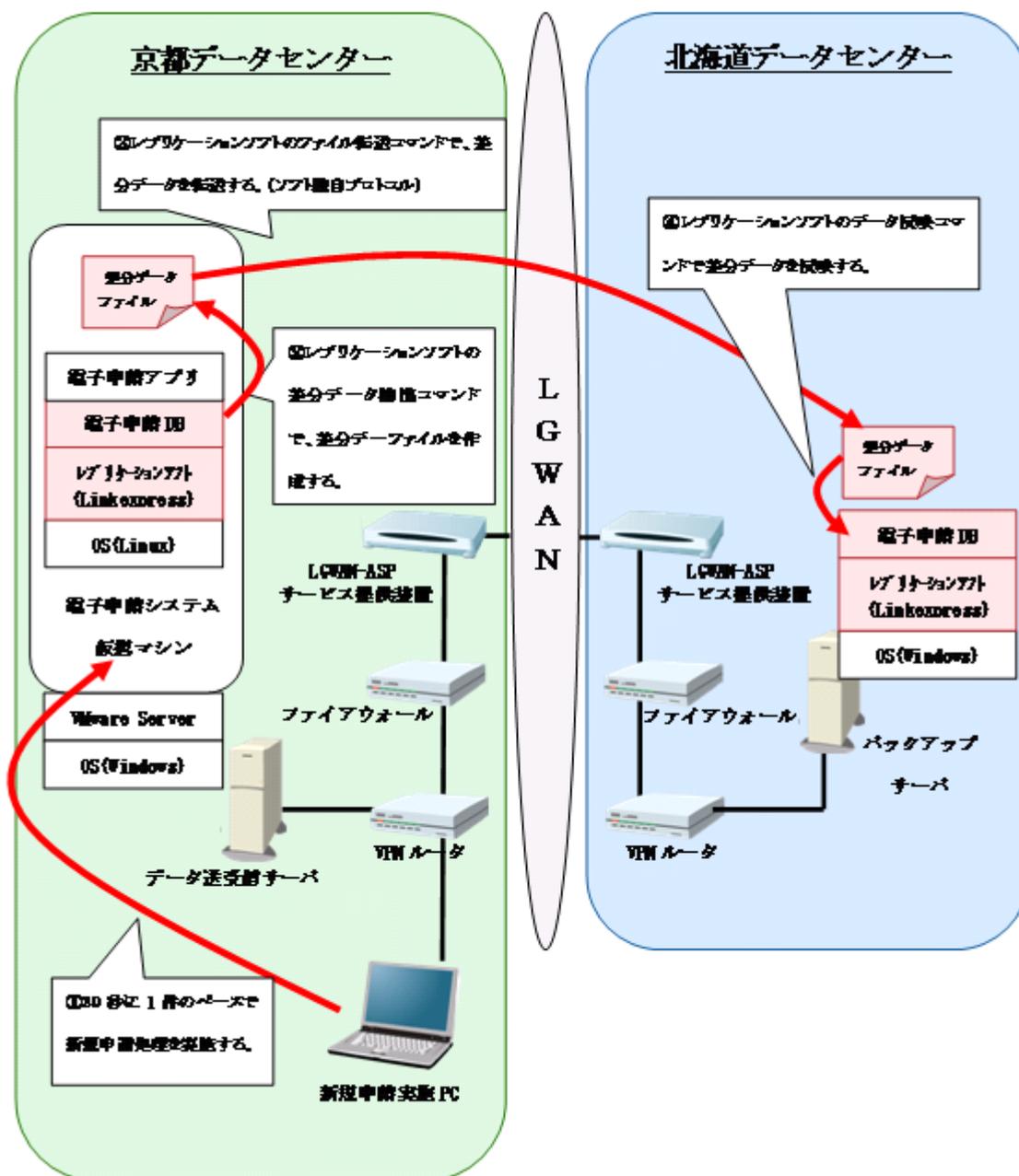


図 4-29 レプリケーション実証イメージ

【目的】

- ① LGWAN経由でオフサイトバックアップ・リストアが正常に実施できるかの実証。
- ② オンサイトバックアップ・リストア及びオフサイトバックアップ・リストアで要した時間を比較しオフサイトバックアップ・リストアの実現性及び有効性の実証。
- ③ LGWAN経由でレプリケーションが正常に実施できるかの実証。
- ④ レプリケーション時のサーバ負荷及び処理時間から見たレプリケーシ

ョンの実現性及び有効性の実証。

B. 実証の内容

a. オフサイトバックアップ・リストア前提条件

A) オフサイトバックアップ対象データベースについて

オフサイトバックアップ対象データベースは、「オンサイトバックアップ」実証実験で使用した電子申請システム仮想マシンの電子申請データベースを対象とした。データ登録件数については、オンサイトバックアップ時と同等である。

B) データベースバックアップ・リストア方法について

電子申請データベースのバックアップデータの作成方法及びリストア方法は、「オンサイトバックアップ」実証実験時と同じ「SymfoWARE Server」のexportコマンド（rdbunl）及びimportコマンド（rdbsloder）で実施する。

C) 京都データセンター北海道データセンター間のネットワークについて

京都データセンター北海道データセンター間は図 4-30 に示すネットワーク構成となっており、両データセンター間はVPNでトンネリングされている。

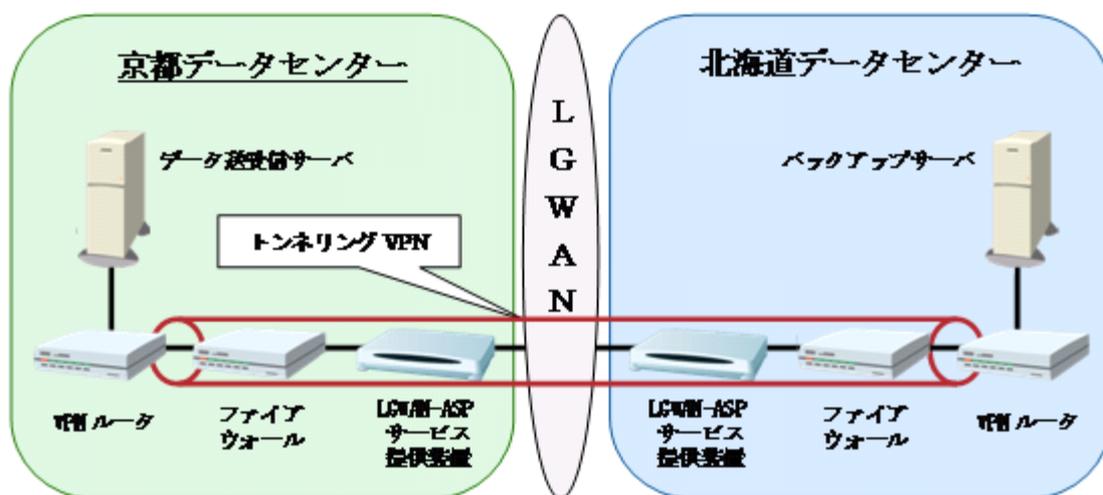


図 4-30 ネットワーク構成図

D) バックアップデータ転送方法について

バックアップデータ転送方法には、FTPやCIFSやファイル転送ソ

フトの利用など様々な方法があるが、本実証実験はFTPで実施する。バックアップ先となるバックアップサーバはIISでFTPサーバを構築し、バックアップ元となる電子申請システム仮想マシンは、標準添付されているftpコマンドを使用する。バックアップサーバへのバックアップ時にはftpコマンドのputを使用し、電子申請システム仮想マシンへのリカバリ時にはftpコマンドのgetを使用する。バックアップデータの転送イメージを図4-31に示す。

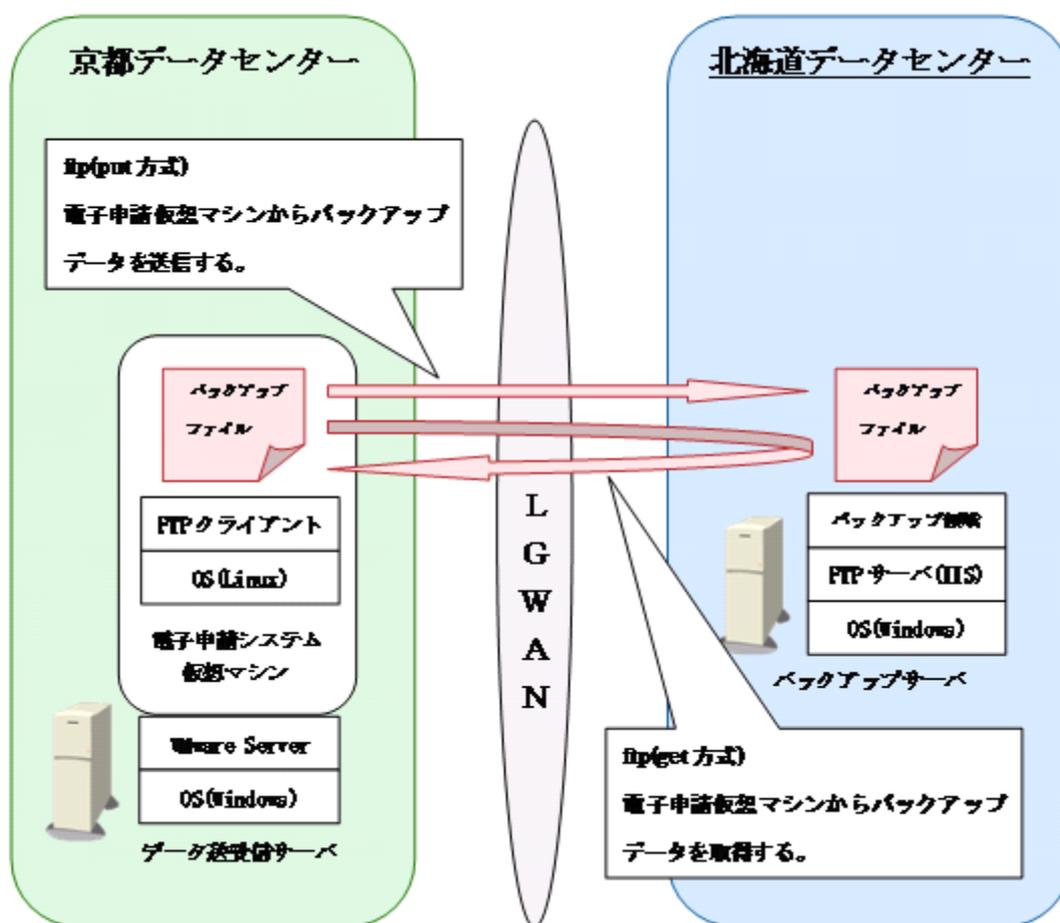


図 4-31 バックアップデータ伝送イメージ

b. レプリケーション前提条件

A) レプリケーション対象データベースについて

レプリケーション対象データベースは、「オンサイトバックアップ」実証実験で使用した電子申請システム仮想マシンの電子申請データベースとバックアップサーバに構築した電子申請データベースを対象とする。バックアップサーバの電子申請データベースは電子申請システム仮想マシンの電子申請データベースと同じテーブル構造で構築する。

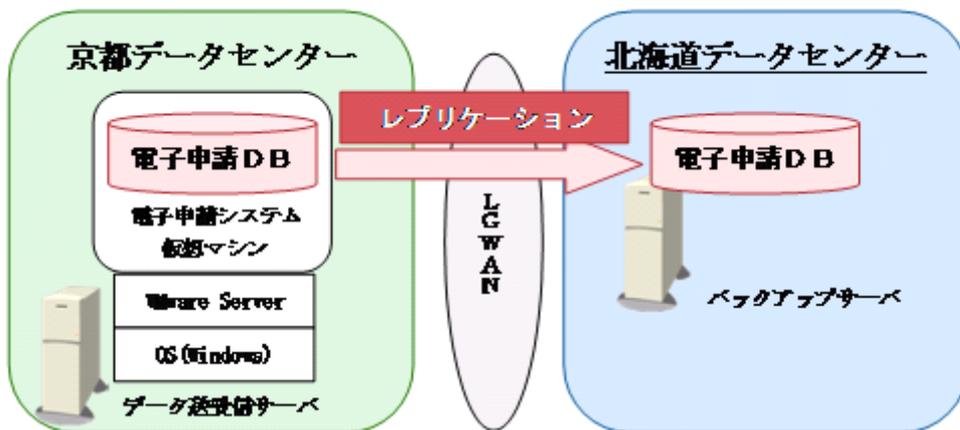


図 4-32 レプリケーション対象データベース構成図

B) レプリケーション方法について

レプリケーションを実施する電子申請システム仮想マシン及びバックアップサーバの電子申請データベースは「SymfoWARE Server」で構築されている。これらのデータベースをレプリケーションする方法として「SymfoWARE Server」のデータベースレプリケーションが可能なソフトウェア「Linkexpress」と「Linkexpress Replication Option」を使用する。ソフトウェア構成図を図 4-33 に示す。

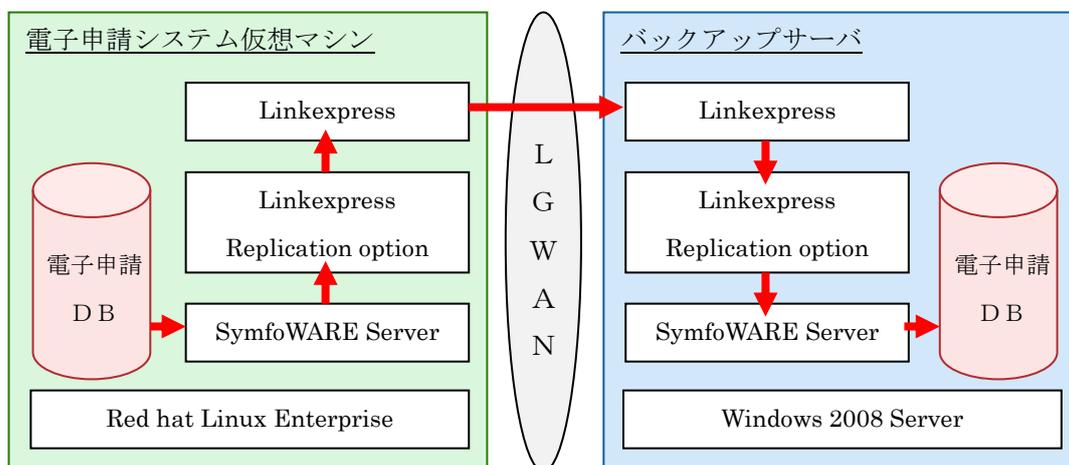


図 4-33 レプリケーションソフトウェア構成図

本実証実験で採用した「Linkexpress」は、マルチプラットフォーム間を高信頼かつ簡単につなぐデータ連携ソフトウェアで以下の特長を持つ。

- ファイル転送機能
  - ・ オープンなプロトコルである「FTP/HTTP/HTTPS」をサポートしLinkexpressを搭載していない既存のFTPサー

バやHTTPサーバへのファイル転送を実現する。また、FTPの信頼性問題を解消した独自の高信頼性プロトコル「FTP+」でのファイル転送も可能で、複数ファイルの一括転送、4GB以上のデータ転送、コード変換、データ圧縮、データ暗号化、ファイル転送失敗時のリトライ等の機能を有する。

- アプリケーション（ジョブ）連携機能
  - ・ 分散システム間で業務プログラムの実行と結果を連携相手に通知できる。業務プログラムの実行はファイル転送と独立して起動できる。応用例として自動スケジュール機能と連携し、ジョブスケジュール的な使用も可能である。また、利用者プログラム間で簡易的なメッセージを交換することができるので、分散システムの利用者プログラム間で連携（同期）をとることができる。
  - ・ API（利用者プログラムインターフェース）を利用することでカスタマイズもできる。この機能により、業務プログラムから直接ファイル転送を実行することや、分散システムの利用者プログラム間で連携（同期）等が可能である。
- 運用管理機能
  - ・ 業務の監視はGUI画面で操作でき、実行待ち（未処理）、実行中、正常/異常完了などの業務のステータスをシグナル表示する。また業務等の実行時に履歴がログファイルに記録される。
  - ・ ファイル転送や業務プログラムの実行を、自動スケジュールで起動することができる。

本実証実験で採用した「Linkexpress Replication Option」はLinkexpressにレプリケーション機能を追加するソフトウェアで以下の特長を持つ。

- ・ レプリケーション対象となるすべてのデータを一括してレプリケートする完全複写方式と更新されたデータだけを取り出してレプリケートする高速複写（差分複写）方式をサポートする。
- ・ OracleやSQL ServerなどのデータベースやAIM/DBやVSAMといったデータベース以外のリソースとの異種データベースレプリケーションが可能である。

「Linkexpress」と「Linkexpress Replication Option」を用いたレプリケーションイメージを図4-34に示す。

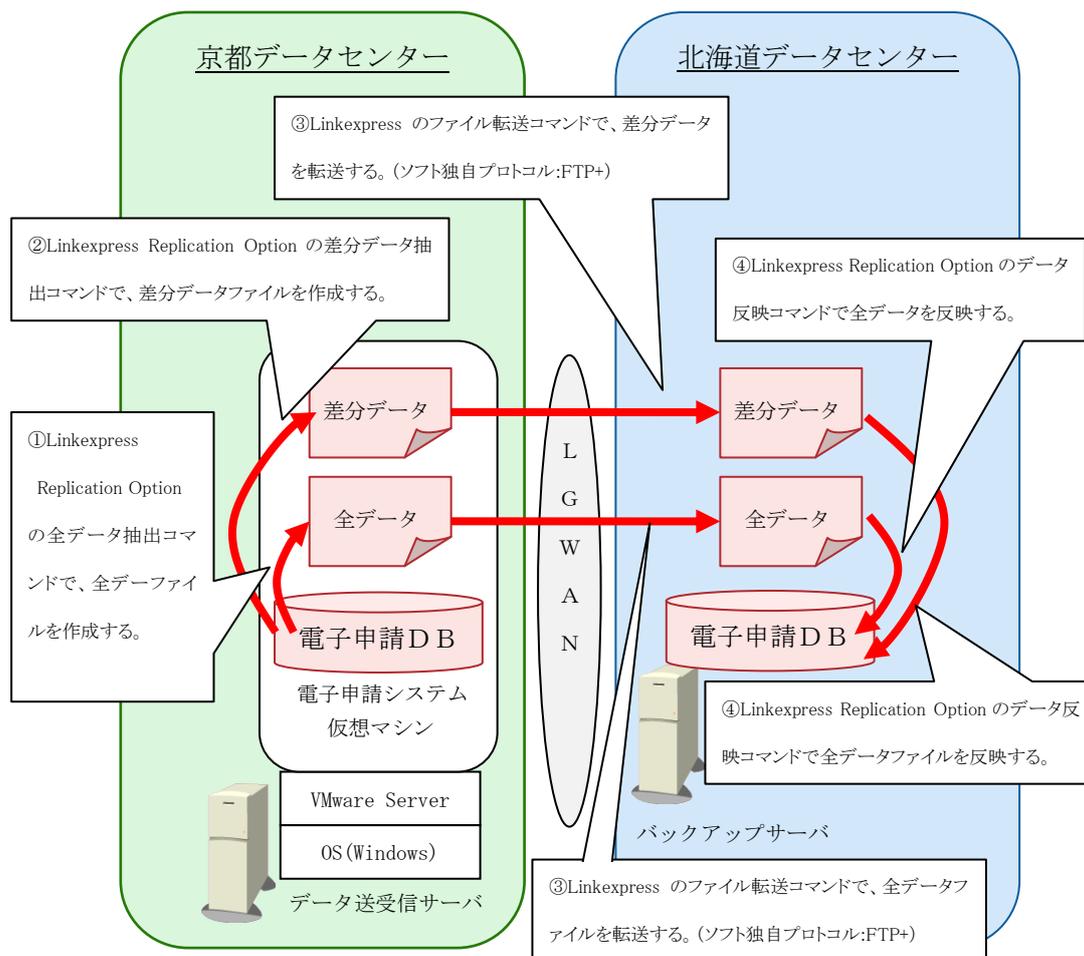


図 4-34 Linkexpressを用いたレプリケーションイメージ

C) 京都データセンター北海道データセンター間のネットワークについて  
 京都データセンター北海道データセンター間はオフサイトバックアップ・リカバリ実証実験時と同じネットワーク構成となっており、両データセンター間はVPNでトンネリングされている。

D) 電子申請データベース更新方法について

レプリケーションを実施するにあたり、電子申請システム仮想マシンの電子申請データベースに対してデータ追加等の更新処理を行う必要がある。更新方法としてPCから電子申請システム仮想マシンに対して申請処理を30秒に1件のペースで実施し、電子申請データベースに対してデータ追加を行う。また、申請処理は手動で実施するのではなく、Webアプリケーションテストソフト「e-TEST suite」を使用して自動的に申請処理を行う。申請処理で使用する手続きは京都府電子申請システムの動作確認用の手続きを使用する。

「e-TEST suite」はWebアプリケーションの負荷テスト

ツールe-Load、リグレーション自動テストツールe-Tester、テスト要件から不具合までテストに関わる情報を一元管理するe-Manager Enterpriseで構成されており、本実証実験では、リグレーション自動テストツールe-Testerを使用する。

#### E) レプリケーション実施間隔について

本実証実験では、レプリケーション時に作成される差分ファイルのサイズを変更して実証を行うため、PCから電子申請システム仮想マシンに対して申請処理を30秒に1件のペースで実施し始めてから以下の時間経過後にレプリケーションを実施する。

- 5分
- 2時間
- 12時間

#### F) サーバ性能情報取得方法について

Windowsマシン(データ送受信サーバ及びバックアップサーバ)の性能情報は管理ツールの一つである「信頼性とパフォーマンスモニタ」を使用して取得する。

Linuxマシン(電子申請システム仮想マシン)の性能情報はOS標準コマンドtop、sar、iostat、freeで取得する。

取得対象はCPU使用率、メモリ使用率、ディスク負荷率とし、取得間隔は5秒間隔で取得する。

#### c. 実施手順

前提条件を踏まえ、本実証実験は図 4-35 に示す作業フローに従い実施する。

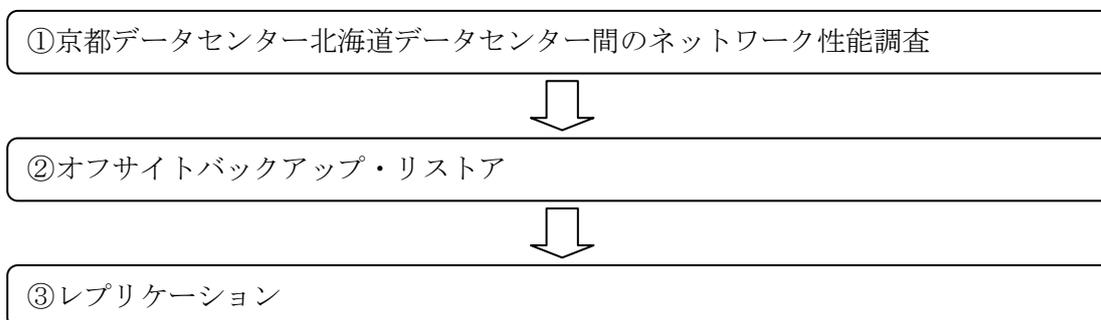


図 4-35 実証実験作業フロー

以下に各作業手順について説明する。

- A) 京都データセンター北海道データセンター間のネットワーク性能調査  
 オフサイトバックアップ・リストア及びレプリケーション実証実験の事前準備として、京都データセンター北海道データセンター間で最も効率よくデータの送受信が行えるMTUの調査とGB単位のファイル送受信が可能か調査する。  
 調査方法を表 4-29 に示す。

表 4-29 ネットワーク性能調査実施手順

手順	作業概要	作業内容
1	Ping コマンドによる MTU 最適値調査	バックアップサーバに対して、データサイズを変えて Ping コマンドを実行し、正常送信可能な最大データサイズを調査する。そのデータサイズを元に最適な MTU 値を算出する。
2	各 MTU 値のファイル転送速度測定	データ送受信サーバ及びバックアップサーバの MTU 値を変えて 150MB のファイル転送 (Windows ファイル共有の copy コマンド) を実施し、手順 1 で算出した MTU 値が最もファイル転送効率が良いか確認する。
3	GB 単位ファイル転送速度測定	データ送受信サーバからバックアップサーバに対して GB 単位のファイル転送 (Windows ファイル共有の copy コマンド) を実施し、送信可能か確認する。

- B) オフサイトバックアップ・リストア

前提条件を踏まえ、本実証実験では表 4-30 に示す作業手順でオフサイトバックアップ・リストアを実施する。また、各作業の処理時間、ファイル転送速度を計測する。

表 4-30 オンサイトバックアップ・リストア実施手順

手順	作業概要	作業内容
1	バックアップデータ作成	電子申請システム仮想マシンでデータベースソフト(SymfoWARE)の export コマンドを実行し、バックアップファイルを作成する。
2	バックアップデータのオフサイトバックアップ	FTP でバックアップファイルをバックアップサーバのバックアップ領域にコピー (put) する。
3	バックアップデータのオフサイトリストア	FTP で、バックアップサーバのバックアップ領域に存在するバックアップファイルを電子申請システム仮想マシンにコピー (get) する。
4	バックアップデータのデータベースリストア	電子申請システム仮想マシンでデータベースソフト(SymfoWARE)の import コマンドを実行し、バックアップファイルをリストアする。

- C) レプリケーション

前提条件を踏まえ、本実証実験では表 4-14 に示す作業手順でレプリケーションを実施する。また、各作業の処理時間、ファイル転送速度及びサーバ性能情報を計測する。

表 4-31 オンサイトバックアップ・リストア実施手順

手順	作業概要	作業内容
1	全DBデータ抽出	電子申請システム仮想マシンでデータベースレプリケーションソフトのDB抽出コマンドを実行し、電子申請データベースの全データを抽出する。
2	全DBデータファイル転送	電子申請システム仮想マシンでデータベースレプリケーションソフトのファイル転送コマンドを実行し、全データファイルをバックアップサーバにコピーする。
3	全DBデータ取込み	バックアップサーバでデータベースレプリケーションソフトの全データ反映コマンドを実行し、電子申請データベースに全データを取り込む。 ※この時点で電子申請システム仮想マシンのデータベースとバックアップサーバのデータベースの同期がとれたことになる。(データ完全一致)
4	新規申請処理実行	電子申請システム仮想マシンに対して、約 30 秒に 1 件の割合で申請処理を実施する。1 件あたりの申請データ (XML ファイル) は 2.4KB で、入力項目が少ない申請手続きと想定している。
5	差分データ抽出	電子申請システム仮想マシンでデータベースレプリケーションソフトの差分データ抽出コマンドを実行し、電子申請データベースの差分データファイルを作成する。 差分データ抽出コマンド実施間隔は 5 分、2 時間、12 時間の 3 パターンで実施する。
6	差分データファイル転送	電子申請システム仮想マシンでデータベースレプリケーションソフトのファイル転送コマンドを実行し、差分データファイルをバックアップサーバにコピーする。
7	差分データ取込み	データベースレプリケーションソフトの差分データ反映コマンドで電子申請データベースに差分データを取り込む。

### C. 実証の結果

#### a. 京都データセンター北海道データセンター間のネットワーク性能調査

##### A) Ping コマンドによる MTU 調査結果

表 4-32 実行コマンド及びその実行結果 (962Byte)

実行コマンド	ping -f -l 962 192.168.xxx.xxx
結果	Pinging 192.168.xxx.xxx with 962 bytes of data: Reply from 192.168. xxx.xxx: bytes=962 time=56ms TTL=128 ←正常応答 Reply from 192.168. xxx.xxx bytes=962 time=51ms TTL=128 ←正常応答 Reply from 192.168. xxx.xxx bytes=962 time=49ms TTL=128 ←正常応答 Reply from 192.168.5 xxx.xxx bytes=962 time=49ms TTL=128 ←正常応答

表 4-33 実行コマンド及びその実行結果 (963Byte)

実行コマンド	ping -f -l 963 192.168.xxx.xxx
結果	Pinging 192.168.xxx.xxx with 963 bytes of data: Packet needs to be fragmented but DF set. ←エラー応答 Packet needs to be fragmented but DF set. ←エラー応答 Packet needs to be fragmented but DF set. ←エラー応答 Packet needs to be fragmented but DF set. ←エラー応答

962Byteのデータは正常に送信できたが、963Byteのデータ送信時はフラグメントが発生した。

上記の結果から京都データセンター北海道データセンター間の最適な

MTU値は以下の計算式より「990」と判断した。

$$\text{MTU} = 962 \text{ (データサイズ)} + 8 \text{ (ICMPヘッダ)} + 20 \text{ (IPヘッダ)} \\ = 990$$

#### B) 各MTU値のファイル転送調査結果

データ送受信サーバ及びバックアップサーバのMTU値を変更し、ファイルサイズが150MBのデータをデータ送受信サーバからバックアップサーバに転送（Windowsファイル共有のcopy）した結果を表4-34に示す。

表 4-34 MTU値別ファイル転送結果

MTU 値	転送時間	転送スピード
400	250.06 秒	617.27KB/秒
600	215.52 秒	716.22KB/秒
990	207.55 秒	743.72KB/秒
1100	240.73 秒	641.19KB/秒

表 4-34 の結果より、京都データセンター北海道データセンター間の最適なMTU値は「990」と判断した。

#### C) 各ファイルサイズのファイル転送調査結果

データ送受信サーバ及びバックアップサーバのMTU値を990に設定し、ファイルサイズが大きいデータをデータ送受信サーバからバックアップサーバに転送（Windowsファイル共有のcopy）した結果を表4-35に示す。

表 4-35 ファイルサイズ別ファイル転送結果

ファイルサイズ	転送時間	転送スピード
500MB	11 分 37 秒	734.63KB/秒
1GB	23 分 49 秒	733.50KB/秒
5GB	2 時間 00 分 32 秒	724.85KB/秒
10GB	3 時間 59 分 37 秒	729.39KB/秒

表 4-35 の結果より、ファイルサイズ10GBまではファイル転送することができることを確認できた。限界値については不明である。

#### b. オフサイトバックアップ・リストア

今回実施したオフサイトバックアップ・リストアの結果及び各作業の所要時間を表4-36に示す。

表 4-36 オフサイトバックアップ・リストアの所要時間

手順	作業概要	作業内容	結果	所要時間
1	バックアップデータ作成	電子申請システム仮想マシンでデータベースソフト(SymfoWARE)の export コマンドを実行し、バックアップファイルを作成する。	OK	6分 50秒
2	バックアップデータのオフサイトバックアップ	FTP でバックアップファイルをバックアップサーバのバックアップ領域にコピー(put)する。	OK	30分 12秒
3	バックアップデータのオフサイトリストア	FTP で、バックアップサーバのバックアップ領域に存在するバックアップファイルを電子申請システム仮想マシンにコピー(get)する。	OK	34分 09秒
4	バックアップデータのデータベースリストア	電子申請システム仮想マシンでデータベースソフト(SymfoWARE)の import コマンドを実行し、バックアップファイルをリストアする。	OK	28分 19秒
合計				99分 30秒

オフサイトバックアップ・リストア作業手順1～4実施時に出入力されたバックアップデータのファイルサイズを表 4-37 に示す。

表 4-37 バックアップファイルサイズ

テーブル	手順 1	手順 2	手順 3	手順 4
	ファイルサイズ	ファイルサイズ	ファイルサイズ	ファイルサイズ
マスタ系テーブル 1	84KB	84KB	84KB	84KB
マスタ系テーブル 2	212KB	212KB	212KB	212KB
マスタ系テーブル 3	5,630KB	5,630KB	5,630KB	5,630KB
マスタ系テーブル 4	2,739KB	2,739KB	2,739KB	2,739KB
マスタ系テーブル 5	12KB	12KB	12KB	12KB
マスタ系テーブル 6	2,536KB	2,536KB	2,536KB	2,536KB
マスタ系テーブル 7	2KB	2KB	2KB	2KB
マスタ系テーブル 8	316KB	316KB	316KB	316KB
マスタ系テーブル 9	4KB	4KB	4KB	4KB
マスタ系テーブル 10	33,746KB	33,746KB	33,746KB	33,746KB
マスタ系テーブル 11	19,179KB	19,179KB	19,179KB	19,179KB
マスタ系テーブル 12	4,302KB	4,302KB	4,302KB	4,302KB
マスタ系テーブル 13	3KB	3KB	3KB	3KB
マスタ系テーブル 14	10KB	10KB	10KB	10KB
マスタ系テーブル 15	60KB	60KB	60KB	60KB
マスタ系テーブル 16	208KB	208KB	208KB	208KB
マスタ系テーブル 17	256KB	256KB	256KB	256KB
マスタ系テーブル 18	4,608KB	4,608KB	4,608KB	4,608KB
マスタ系テーブル 19	1,066KB	1,066KB	1,066KB	1,066KB
マスタ系テーブル 20	55KB	55KB	55KB	55KB
マスタ系テーブル 21	5,344KB	5,344KB	5,344KB	5,344KB

テーブル	手順 1	手順 2	手順 3	手順 4
	ファイルサイズ	ファイルサイズ	ファイルサイズ	ファイルサイズ
マスタ系テーブル 22	6KB	6KB	6KB	6KB
マスタ系テーブル 23	517KB	517KB	517KB	517KB
マスタ系テーブル 24	101KB	101KB	101KB	101KB
マスタ系テーブル 25	10KB	10KB	10KB	10KB
トランザクション系テーブル 1	1,200KB	1,200KB	1,200KB	1,200KB
トランザクション系テーブル 2	528,659KB	528,659KB	528,659KB	528,659KB
トランザクション系テーブル 3	358,363KB	358,363KB	358,363KB	358,363KB
トランザクション系テーブル 4	2KB	2KB	2KB	2KB
トランザクション系テーブル 5	3,373KB	3,373KB	3,373KB	3,373KB
トランザクション系テーブル 6	1,968KB	1,968KB	1,968KB	1,968KB
トランザクション系テーブル 7	2,432KB	2,432KB	2,432KB	2,432KB
トランザクション系テーブル 8	2,686KB	2,686KB	2,686KB	2,686KB
トランザクション系テーブル 9	21KB	21KB	21KB	21KB
トランザクション系テーブル 10	561KB	561KB	561KB	561KB
トランザクション系テーブル 11	18KB	18KB	18KB	18KB
トランザクション系テーブル 12	8KB	8KB	8KB	8KB
トランザクション系テーブル 13	2KB	2KB	2KB	2KB
トランザクション系テーブル 14	2KB	2KB	2KB	2KB
トランザクション系テーブル 15	20,310KB	20,310KB	20,310KB	20,310KB
トランザクション系テーブル 16	1,134KB	1,134KB	1,134KB	1,134KB
トランザクション系テーブル 17	9,062KB	9,062KB	9,062KB	9,062KB
トランザクション系テーブル 18	2,062KB	2,062KB	2,062KB	2,062KB
トランザクション系テーブル 19	303,065KB	303,065KB	303,065KB	303,065KB
トランザクション系テーブル 20	1KB	1KB	1KB	1KB
トランザクション系テーブル 21	1KB	1KB	1KB	1KB
トランザクション系テーブル 22	2KB	2KB	2KB	2KB
トランザクション系テーブル 23	2KB	2KB	2KB	2KB
トランザクション系テーブル 24	2KB	2KB	2KB	2KB
合計	1,315,942KB	1,315,942KB	1,315,942KB	1,315,942KB

オフサイトバックアップ・リストア作業前及び手順 4 実施後のデータ登録数を表 4-38 に示す。

表 4-38 データベースデータ登録数

テーブル	作業前	手順 4 実施後
	データ登録数	データ登録数
マスタ系テーブル 1	381	381
マスタ系テーブル 2	714	714
マスタ系テーブル 3	46,465	46,465
マスタ系テーブル 4	19,493	19,493
マスタ系テーブル 5	159	159

テーブル	作業前	手順 4 実施後
	データ登録数	データ登録数
マスタ系テーブル 6	25,519	25,519
マスタ系テーブル 7	2	2
マスタ系テーブル 8	2,870	2,870
マスタ系テーブル 9	35	35
マスタ系テーブル 10	392	392
マスタ系テーブル 11	121,684	121,684
マスタ系テーブル 12	21,331	21,331
マスタ系テーブル 13	9	9
マスタ系テーブル 14	61	61
マスタ系テーブル 15	268	268
マスタ系テーブル 16	1,438	1,438
マスタ系テーブル 17	1,434	1,434
マスタ系テーブル 18	11,035	11,035
マスタ系テーブル 19	8,207	8,207
マスタ系テーブル 20	610	610
マスタ系テーブル 21	30,567	30,567
マスタ系テーブル 22	61	61
マスタ系テーブル 23	862	862
マスタ系テーブル 24	648	648
マスタ系テーブル 25	243	243
トランザクション系テーブル 1	2,107	2,107
トランザクション系テーブル 2	5,320	5,320
トランザクション系テーブル 3	2,209	2,209
トランザクション系テーブル 4	1	1
トランザクション系テーブル 5	9,930	9,930
トランザクション系テーブル 6	12,037	12,037
トランザクション系テーブル 7	2,544	2,544
トランザクション系テーブル 8	17,681	17,681
トランザクション系テーブル 9	64	64
トランザクション系テーブル 10	128	128
トランザクション系テーブル 11	162	162
トランザクション系テーブル 12	87	87
トランザクション系テーブル 13	13	13
トランザクション系テーブル 14	11	11
トランザクション系テーブル 15	34,544	34,544
トランザクション系テーブル 16	12	12
トランザクション系テーブル 17	49	49
トランザクション系テーブル 18	2,709	2,709
トランザクション系テーブル 19	466	466
トランザクション系テーブル 20	1	1
トランザクション系テーブル 21	3	3

テーブル	作業前	手順 4 実施後
	データ登録数	データ登録数
トランザクション系テーブル 22	2	2
トランザクション系テーブル 23	1	1
トランザクション系テーブル 24	1	1

オフサイトバックアップ・リストア手順2実施時のFTPファイル転送速度及びオフサイトバックアップ・リストア手順3実施時のFTPファイル転送速度を表 4-39 に示す。

表 4-39 FTPファイル転送速度結果

転送方向	転送速度
電子申請システム仮想マシン→バックアップサーバ	726.23KB/秒
バックアップサーバ→電子申請システム仮想マシン	642.23KB/秒

c. レプリケーション

今回実施したレプリケーションの結果及び所要時間を表 4-40 に示す。

表 4-40 レプリケーションの結果と所要時間

手順	作業対象 機器	作業内容	結果	所要時間
1	全DBデータ抽出	データベースレプリケーションソフトのDB抽出コマンドを使用し、電子申請データベースの全データを抽出する。	OK	5分50秒
2	全DBデータファイル転送	データベースレプリケーションソフトのファイル転送コマンドで全データファイルをバックアップサーバにコピーする。	OK	29分55秒
3	全DBデータ取込み	データベースレプリケーションソフトの全データ反映コマンドで電子申請データベースに全データを取り込む。 ※この時点で電子申請システム仮想マシンのデータベースとバックアップサーバのデータベースの同期がとれたことになる。(データ完全一致)	OK	29秒
4	新規申請処理実行	電子申請システム仮想マシンに対して、約30秒に1件の割合で申請処理を実施する。	OK	-
5	差分データ抽出	データベースレプリケーションソフトの差分データ抽出コマンドで電子申請データベースの差分データファイルを作成する。差分データ抽出コマンド実施間隔は5分、2時間、12時間の3パターンで実施する。	OK	5分:3秒 2時間:8秒 12時間:15秒
6	差分データファイル転送	データベースレプリケーションソフトのファイル転送コマンドで差分データファイルをバックアップサーバにコピーする。	OK	5分:6秒 2時間:10秒 12時間:25秒
7	差分データ取込み	データベースレプリケーションソフトの差分データ反映コマンドで電子申請データベースに差分データを取り込む。	OK	5分:1秒 2時間:1秒 12時間:1秒

レプリケーション手順1実施時に出力されたバックアップデータのファイルサイズ及びレコード数を表 4-41 に示す。

表 4-41 バックアップファイルサイズ及びレコード数

テーブル	ファイルサイズ	レコード数
マスタ系テーブル 1	84KB	381
マスタ系テーブル 2	212KB	714
マスタ系テーブル 3	5,630KB	46,465
マスタ系テーブル 4	2,739KB	19,493
マスタ系テーブル 5	12KB	159
マスタ系テーブル 6	2,536KB	25,519
マスタ系テーブル 7	2KB	2
マスタ系テーブル 8	316KB	2,870
マスタ系テーブル 9	4KB	35
マスタ系テーブル 10	33,746KB	392
マスタ系テーブル 11	19,179KB	121,684
マスタ系テーブル 12	4,302KB	21,331
マスタ系テーブル 13	3KB	9
マスタ系テーブル 14	10KB	61
マスタ系テーブル 15	60KB	268
マスタ系テーブル 16	208KB	1,438
マスタ系テーブル 17	256KB	1,434
マスタ系テーブル 18	4,608KB	11,035
マスタ系テーブル 19	1,066KB	8,207
マスタ系テーブル 20	55KB	610
マスタ系テーブル 21	5,344KB	30,567
マスタ系テーブル 22	6KB	61
マスタ系テーブル 23	517KB	862
マスタ系テーブル 24	101KB	648
マスタ系テーブル 25	10KB	243
トランザクション系テーブル 1	1,200KB	2,107
トランザクション系テーブル 2	528,659KB	5,320
トランザクション系テーブル 3	358,363KB	2,209
トランザクション系テーブル 4	2KB	1
トランザクション系テーブル 5	3,373KB	9,930
トランザクション系テーブル 6	1,968KB	12,037
トランザクション系テーブル 7	2,432KB	2,544
トランザクション系テーブル 8	2,686KB	17,681
トランザクション系テーブル 9	21KB	64
トランザクション系テーブル 10	561KB	128
トランザクション系テーブル 11	18KB	162
トランザクション系テーブル 12	8KB	87
トランザクション系テーブル 13	2KB	13
トランザクション系テーブル 14	2KB	11
トランザクション系テーブル 15	20,310KB	34,544
トランザクション系テーブル 16	1,134KB	12

テーブル	ファイルサイズ	レコード数
トランザクション系テーブル 17	9,062KB	49
トランザクション系テーブル 18	2,062KB	2,709
トランザクション系テーブル 19	303,065KB	466
トランザクション系テーブル 20	1KB	1
トランザクション系テーブル 21	1KB	3
トランザクション系テーブル 22	2KB	2
トランザクション系テーブル 23	2KB	1
トランザクション系テーブル 24	2KB	1
合計	1,315,942KB	

レプリケーション手順5実施時に出力された差分データのファイルサイズ（5分、2時間、12時間の3パターン）を表 4-42 に示す。差分データは更新のあったテーブルのみ作成される。

表 4-42 パターン毎の差分データファイルサイズ

テーブル	ファイルサイズ		
	5分	2時間	12時間
トランザクション系テーブル 1	5KB	109KB	670KB
トランザクション系テーブル 2	22KB	586KB	3,589KB
トランザクション系テーブル 3	23KB	598KB	3,701KB
トランザクション系テーブル 5	8KB	211KB	1,297KB
トランザクション系テーブル 6	2KB	45KB	277KB
トランザクション系テーブル 7	3KB	62KB	376KB
トランザクション系テーブル 8	10KB	262KB	1,603KB
合計	73KB	1,873KB	11,513KB

レプリケーション手順2及び6実施時のFTP+（ソフト独自プロトコル）ファイル転送速度を表 4-43 に示す。

表 4-43 FTP+（ソフト独自プロトコル）ファイル転送速度結果

パターン(転送ファイルサイズ)	転送速度
全データ転送(1,315,942KB)	733.15KB/秒
5分差分データ(73KB)	12.16KB/秒
2時間差分データ(1,873KB)	185.30KB/秒
12時間差分データ(11,513KB)	460.52KB/秒

レプリケーション実施時のサーバの性能情報を表 4-44 に示す。

表 4-44 レプリケーション実施時のサーバ性能情報

性能情報取得 タイミング	対象機器	CPU 使用率		メモリ 使用率		ディスク 負荷率	
		平均	最大	平均	最大	平均	最大
30秒に1件のペースで申請時	電子申請システム仮想マシン	27%	72%	57%	58%	3%	30%
	データ送受信サーバ	2%	5%	41%	43%	15%	22%
5分差分データ取得時	電子申請システム仮想マシン	28%		59%		4%	
	データ送受信サーバ	4%		59%		38%	
2時間差分データ取得時	電子申請システム仮想マシン	57%		57%		4%	
	データ送受信サーバ	4%		58%		38%	
12時間差分データ取得時	電子申請システム仮想マシン	57%		57%		4%	
	データ送受信サーバ	3%		56%		38%	

※ 5分、2時間、12時間差分データ取得処理はすべて15秒以内で完了し性能を取得時のみデータしかないため、取得したデータのみ記載している。

#### D. 結果の考察

##### a. オフサイトバックアップ・リストア

オンサイトバックアップ・リストアとオフサイトバックアップ・リストアの所要時間比較表を表 4-45 に示す。

表 4-45 バックアップ・リストアの所要時間比較表

項番	オンサイトバックアップ・リストア		オフサイトバックアップ・リストア	
	手順	所要時間	手順	所要時間
1	データベースソフト(SymfoWARE)のダンプコマンドで、バックアップファイルをDB退避領域に作成する。	6分56秒	データベースソフト(SymfoWARE)の export コマンドで、バックアップファイルを作成する。	6分50秒
2	バックアップファイルをテープライブラリ装置のテープにバックアップする。	7分30秒	FTP でバックアップファイルをバックアップサーバのバックアップ領域にコピー(put)する。	30分12秒
3	テープのバックアップデータをDB退避領域にリストアする。	2分56秒	FTP で、バックアップサーバのバックアップ領域に存在するバックアップファイルを電子申請システム仮想マシンにコピー(get)する。	34分09秒
4	データベースソフト(SymfoWARE)のリストアコマンドで、バックアップファイルをDBにリストアする。	29分39秒	データベースソフト(SymfoWARE)の import コマンドで、バックアップファイルをリストアする。	28分19秒
	合計	47分01秒	合計	99分30秒

表 4-45 の結果からバックアップ時間及びリストア時間ともにオンサイトバックアップの方が処理時間は短い。今回実証実験で使用したデータベースのフルバックアップデータ容量は1.3GB程度だったので、バックアップ・リカバリに要する時間は約2倍であったが、大容量のバックアップデータの場合、表 4-28 のテープバックアップリストアスループット値及び表 4-43 のFTPファイル転送速度値からもわかるように、この差はますますひらくと思われる。データの遠地保管の観点でみた場合、オフサイ

トバックアップは有効であるが、障害時のリストア時間を考慮するとオフサイトバックアップのみの運用はリスクが大きい。オンサイトバックアップとオフサイトバックアップの併用運用が有効であると考えられる。

本実証実験の結果では京都データセンターから北海道データセンターへのFTPファイル転送速度は726.23KB/秒で、北海道データセンターから京都データセンターへのFTPファイル転送速度は642.23KB/秒であった。この結果を元にファイルサイズ別ファイル転送予想時間表を作成した。その表を表4-46に示す。

表 4-46 ファイルサイズ別ファイル転送予想時間

ファイルサイズ	転送方向	
	京都→北海道	北海道→京都
10MB	14 秒	16 秒
100MB	2 分 21 秒	2 分 39 秒
500MB	11 分 45 秒	13 分 15 秒
1GB	24 分 03 秒	27 分 08 秒
5GB	2 時間 00 分 15 秒	2 時間 15 分 40 秒
10GB	4 時間 00 分 30 秒	4 時間 31 分 20 秒
100GB	40 時間 05 分 00 秒	42 時間 36 分 40 秒

※ 5GB以上のファイルについてはFTPファイル転送可能か本実証実験では未実証

オフサイトバックアップ・リストア運用を実施するにあたり、SLAの要求項目の一つである目標復旧時間（RTO）を考慮する必要がある。対象となるシステムにより目標復旧時間は異なるが、設定する際は、表4-46の結果を参考にしていきたい。また、バックアップデータはL2/L3を経由して送信されるため、データサイズについて配慮しなければならない。そのため差分・増分などのデータを抽出し、さらに圧縮、重複排除などによりデータサイズの縮小を行う必要があると考えられる。

#### b. レプリケーション

5分間隔レプリケーション処理（差分データ容量：73KB）は10秒、2時間間隔レプリケーション処理（差分データ容量：1,873KB）は19秒、12時間間隔レプリケーション処理（差分データ容量：11,513KB）は41秒で完了した。また、レプリケーション処理時のサーバCPU使用率、メモリ使用率、ディスク負荷率は5分、2時間、12時間とも同値であった。処理時間及びサーバ性能結果から考察すると、L2/L3経由のリアルタイムレプリケーション運用は充分可能だと考えられる。注目していたデータ転送速度（FTP+）についても733.15KB/秒だったで、リアルタイムレプリケーションを実施しても数秒でレプリケーション処理が完了すると思われる。

ただし、データの更新処理が頻発し、その更新データ量も大きいシステ

ムでレプリケーションを実施する場合は、事前に十分な検討及び実証が必要だと考えられる。

レプリケーションに必要と考えられる機能としては、レプリケーション運用中に発生する可能性があるネットワーク障害を考慮し、同期できない場合には同期データをサーバ内に蓄積する機能及び同期データ再送機能が重要だと考えられる。その他にも、データ送信時のネットワーク負荷を考慮したデータ圧縮機能や任意にレプリケーションが実行可能な機能もあればよいと思われる。

(3) アクセス権限情報を含めたバックアップ

ア) 実証の前提

今回の実証実験において、用いた製品と構成図を示す。

表 4-47 実証に用いた製品

実証実験において用いた製品	
・	文書管理 DB : Lotus Notes/Domino 8.5.2
・	Lotus Domino Server 8.5.2(以下 Lotus Domino)
・	Lotus Notes クライアント 8.5.2
・	Lotus Domino Administrator 8.5.2(管理用のみ利用)
・	Lotus Domino Designer 8.5.2(設計変更時のみ利用)
・	基幹系 DB : IBM DB2 9.7(以下 DB2)
・	セキュリティ・クライアント : Self controllable Security Engine 2. 1.2(以下 SSE)

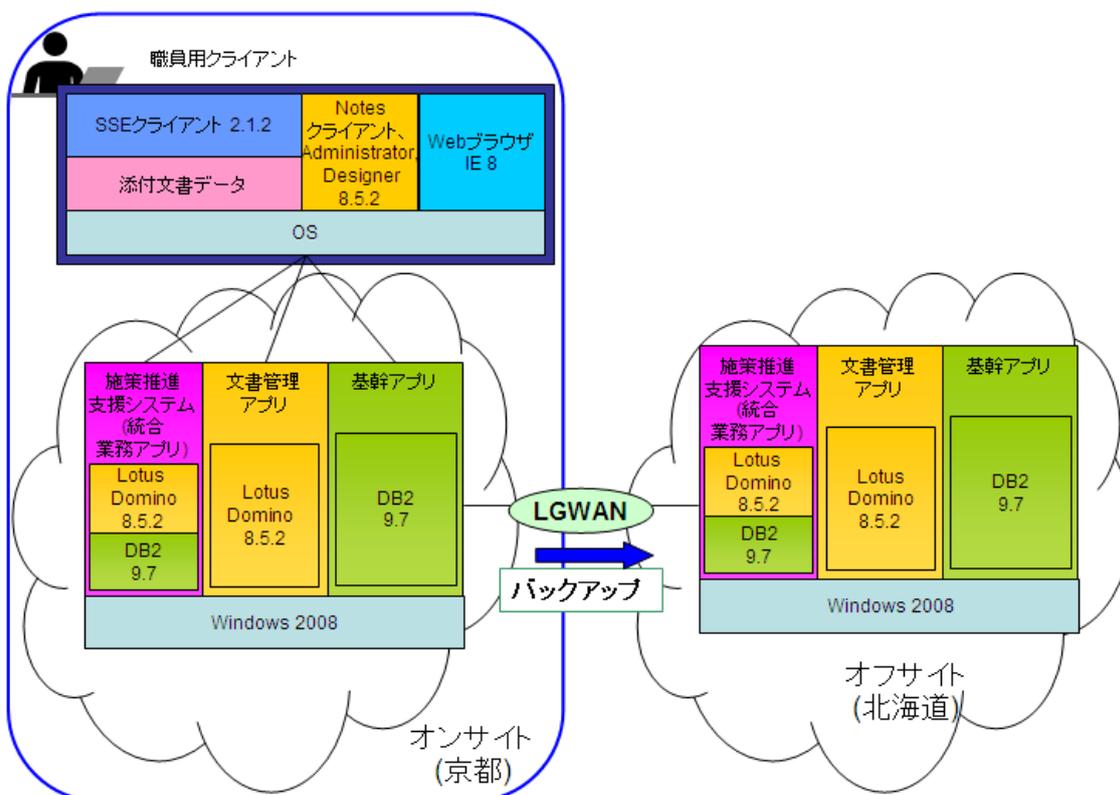


図 4-36 実証実験システム構成

実証実験システム構成の施策推進支援システムは、文書管理DBと基幹系DBの統合アプリケーションである。このシステムは、文書管理DBをインタフェースにデータを入力すると、指定したフィールドに保管されたデータは自動で基幹系DBに保管される設定になっている。ただし、ユーザが文書管理DBにアクセスし、文書を表示した場合、ユーザはどのデータが文書管理DBに保管されており、どのデータが基幹系DBに保管されているかを意識することはない。文書管理DBと基幹系DBの連携イメージについては以下図 4-37 に示す。

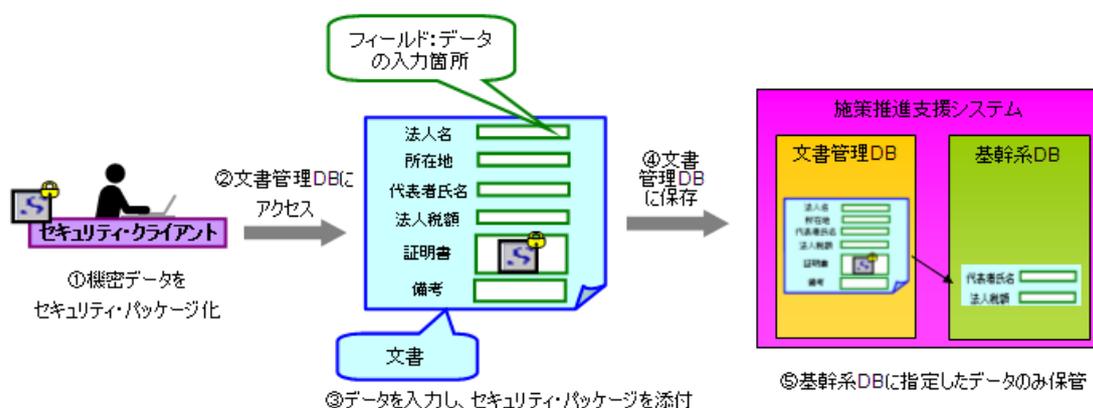


図 4-37 文書管理DBと基幹系DBの連携イメージ

なお、文書管理DBのリアルタイムバックアップにはDominoクラスタ機能を利用する。Dominoクラスタ機能とは、Notes/Dominoの複数のデータベースの文書及び設計要素を「データ更新時」に複製をとり、同期をする機能である。Dominoクラスタは複数台にて構築することもできるため、1台はバックアップ用途で、ユーザからのアクセスなし、その他のサーバはユーザからのアクセスを可能にし、負荷分散用途で利用するなどの運用も可能である。

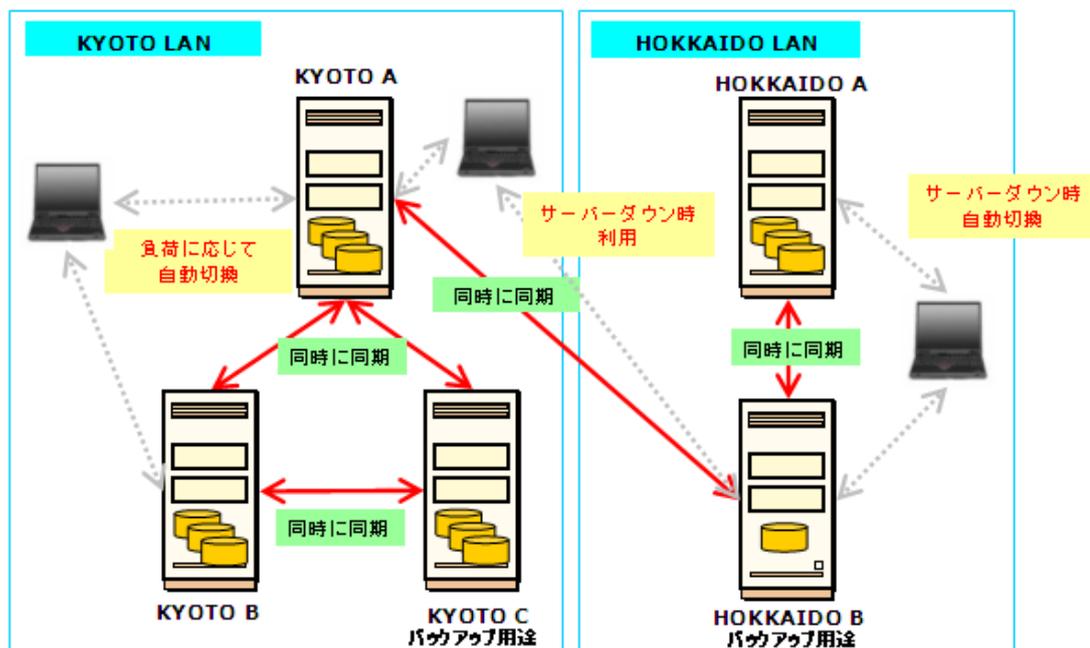


図 4-38 Domino クラスターの一般構成

基幹系データベースの随時リモートバックアップにはDB2 High Availability Disaster Recovery (HADR) 機能を用いる。HADRは稼働系のデータベースへの変更を、ネットワークを介して待機系データベースへ伝達する機能である。

HADRには次の3つの同期モードがある。

#### ■ 同期 (Sync)

このモードはトランザクション損失に対する最大限の保護を提供するが、トランザクション応答時間への影響が最大となる。

待機系と稼働系両方にあるディスク上のデータベースのログ(注)にデータ変更が書かれたら、稼働系DBのトランザクション完了とする。

#### ■ 近似同期 (Near sync)

このモードは、トランザクション損失に対する保護を幾分緩くする。その代わりに、同期 (Sync) モードよりもトランザクション応答時間が短くなる。

待機系DBのログ受信バッファにデータ変更が伝達され、稼働系DBのディスク上のデータベースのログにデータ変更が書かれたら、稼働系DBのトランザクション完了とする。

なお、近似同期がデフォルトの設定である。

#### ■ 非同期 (Async)

このモードは稼働系DBの障害時にトランザクション損失の確率が最も高くなる。その代わりに、3種類のモードの中ではトランザクシ

ョン応答時間が最も短くなる。

稼働系DBのTCP/IP層にデータ変更が渡され、応答が返ったら、稼働系DBのトランザクション完了とする。

(注) 上記説明中に記述したログは、データベースへの全ての更新内容を発生順に記録するファイルである。ログの目的は次の2点である。

- データベース障害時における整合性の維持
- 更新処理におけるパフォーマンスの確保

データベースへの更新があった場合、データベース・マネージャーは最初に更新内容をログに書き出し、その後でデータベース本体を更新する。

参照時のパフォーマンスも考慮したデータベースへの書き込みは比較的負荷の高い処理である。通常運用時には、ログに更新内容を書いてから非同期的にデータベース本体への更新内容の最適な状態で書き込むことで、参照時と更新時の両方のパフォーマンスの両立を図っている。

また、データベースサーバに障害が発生した場合には、データベース・マネージャーはログ上に存在するデータベースへの変更が未済の更新データをデータベースに反映し、整合性の取れたデータベースとしてサービスを再開する。

上記の3つの同期モードのそれぞれがどこまでデータ変更の書き込みが終わった時点でトランザクション完了としているかを図 4-39 に示す。

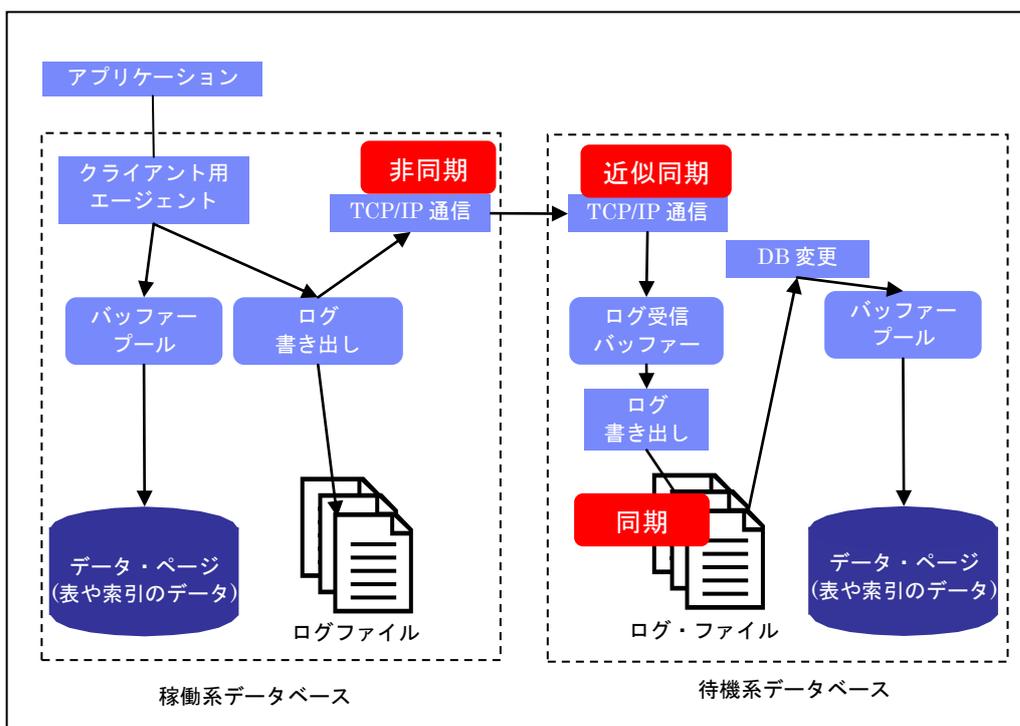


図 4-39 同期モードの違いによるトランザクション完了時点

稼働系DBサーバとバックアップ系DBサーバサーバ、DB2 HADR機能とDB2クライアント接続用のTCP/IPポートでの通信が可能だとしている。本実証実験では次の3ポートを用いた。

- HADRプライマリ用： 50010
- HADRセカンダリー用： 50020
- DB2クライアント用： 50030

#### (4) バックアップ・リストアおよび運用に関する検証

##### ア) 実証の概要・目的

バックアップ・リストアおよび運用に関する検証項目は、次の通りである。

- A) システム全体のバックアップや個別アプリケーションのバックアップなど各自治体の要件対応の為、各自治体データをオフサイトに一括でバックアップ、および個別バックアップできることを確認。また障害時直前まで戻す場合や、災害時にオフサイトのデータを利用して業務を継続することを想定し、バックアップへのデータ反映の即時性を確認
- B) ネットワーク負荷やバックアップ中の業務継続対応の為、オンラインバックアップ、オフラインバックアップおよびスケジュールによるバックアップができることを確認
- C) 障害時復旧の為、オフサイトのバックアップデータを用いたリストアを確認
- D) 各自治体が、オンサイトを利用したデータバックアップ・リストアができることを確認

また、バックアップファイルから機密データが盗まれるような万が一の事態に備えて、データ暗号化の考慮が必要であると考え。今回実証実験の対象となっているアプリケーションにも、特定レコードや文書データを暗号化やアクセス制御するしくみを提供するものがある。

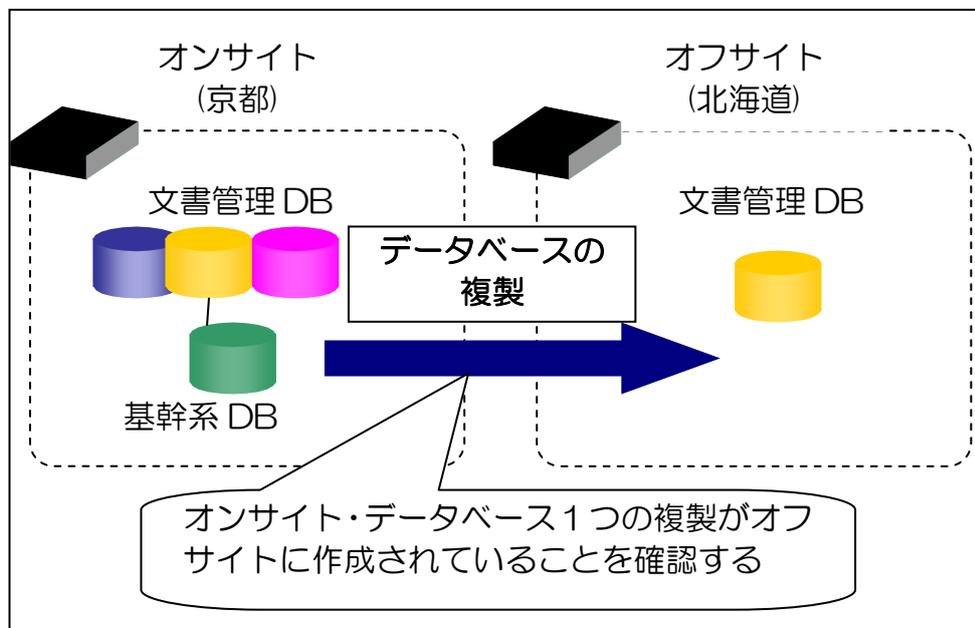
例えば、今回の検証対象の文書管理DBでは、サーバ、データベース、ビュー、文書、フィールドの各単位で、アクセス制御がかけられ、セキュリティを担保することができる。しかし、文書管理ソフトウェアに添付する電子ファイルは複数人に共有されることが多いため、機密ファイルの場合は二次漏洩を防止するために暗号化して添付することが望まれる。また、単なる暗号化では受領者が復号化した後に、USBにコピーされたり、PCのウィルス感染によって二次漏洩したりする危険性があるため、この二次漏洩を考慮したセキュリティのしくみについても今回の実証実験の対象とし、権限データも含めてバックアップを行うことで、リカバリー時に誤って情報漏えいするようなケースを回避することができることを検証した。

イ) 実証の内容

オフサイトを利用したバックアップ・リストア検証について、実証の内容を以下に示す。

A	通常運用時を想定し、自治体クラウドに参加する自治体のデータを、オフサイトにバックアップできることを確認する。
A-1	<p>全データを一括でサーバ稼動中に、バックアップできることを以下の手順により、確認する。</p> <ul style="list-style-type: none"> <li>・ オンサイトに文書管理 DB を 4 つ作成する。</li> <li>・ オンサイトに基幹系 DB を 1 つ作成する。</li> <li>・ オフサイトにも文書管理 DB サーバと基幹系 DB サーバを構築し、それぞれ複製の設定を実施する。</li> <li>・ オンサイトとの通信を確認し、Domino クラスタ、HADR サービスを開始する。</li> <li>・ オフサイトの DB の内容やログから、オンサイトと同様のデータがバックアップされていることを確認する。</li> </ul> <div style="text-align: center; border: 1px solid black; padding: 10px; margin: 10px 0;"> <p>オンサイト (京都)                      オフサイト (北海道)</p> <p>文書管理 DB                      文書管理 DB</p> <p>データベースの複製</p> <p>基幹系 DB                      基幹系 DB</p> <p>オンサイト・データベースの複製がオフサイトに作成されていることを確認する</p> </div>
A-2	<p>データを個別にサーバ稼動中に、バックアップできることを以下の手順にて確認する。 【文書管理 DB】</p> <ul style="list-style-type: none"> <li>・ オンサイトで新規に文書管理 DB を作成し、文書を登録する。</li> <li>・ 作成した文書管理 DB を選択し、オフサイトのサーバへの複製設定を行う。</li> <li>・ 複製を実行する。</li> </ul>

- ・ 新規で作成した文書管理 DB がオフサイトにバックアップされていることを確認する。
- ・ オフサイトの文書管理 DB にアクセスし、オンサイトで登録した文書が登録されていることを確認する。



A-3

データの新規作成、編集、削除、ファイルのアップロードの各操作において、バックアップデータに即時反映できることを以下の手順にて確認する。

【文書管理 DB】

- ・ オンサイトの文書管理 DB に文書を作成する。
- ・ オフサイトの文書管理 DB のバックアップにもデータが追加されていることを確認する。
- ・ 目視およびログよりバックアップデータへの反映時間を確認する。

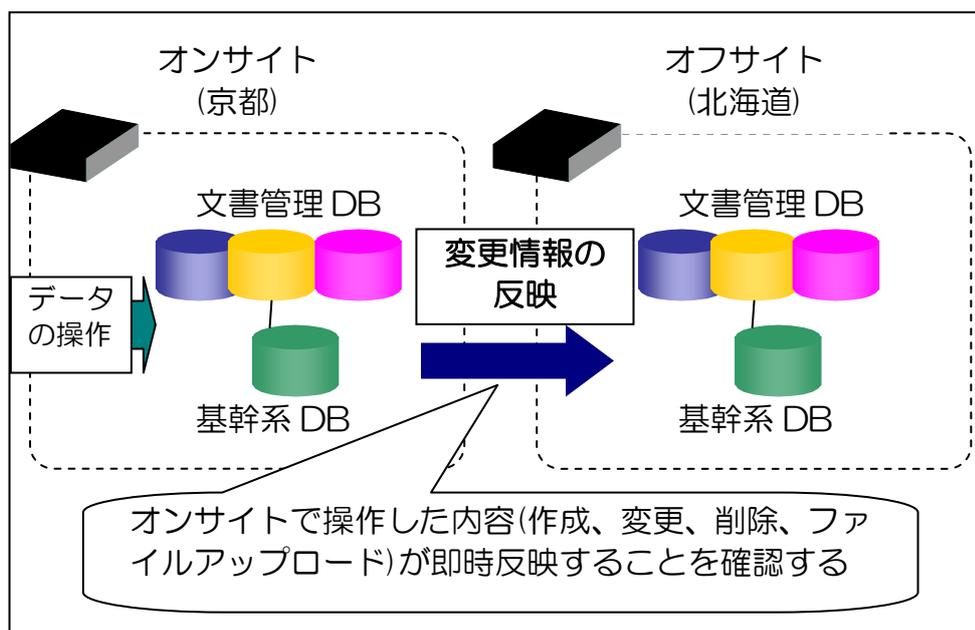
【基幹系 DB】

- ・ 基幹系 DB にデータを挿入する。
- ・ オフサイトの基幹系 DB のバックアップにもデータが挿入されていることを確認する。
- ・ ログよりバックアップデータへの反映時間を確認する。

※上記操作の文書作成および、データ挿入部分を、オンサイトからのデータの編集、削除、ファイルのアップロードに置き換え、それぞれオフサイトのバックアップが即時更新されることを確認する。特に、ファイルアップロードについては、ファイル容量を 1 MB、100MB、1GB と変更し、ファイル容量による反映時間を測定する。

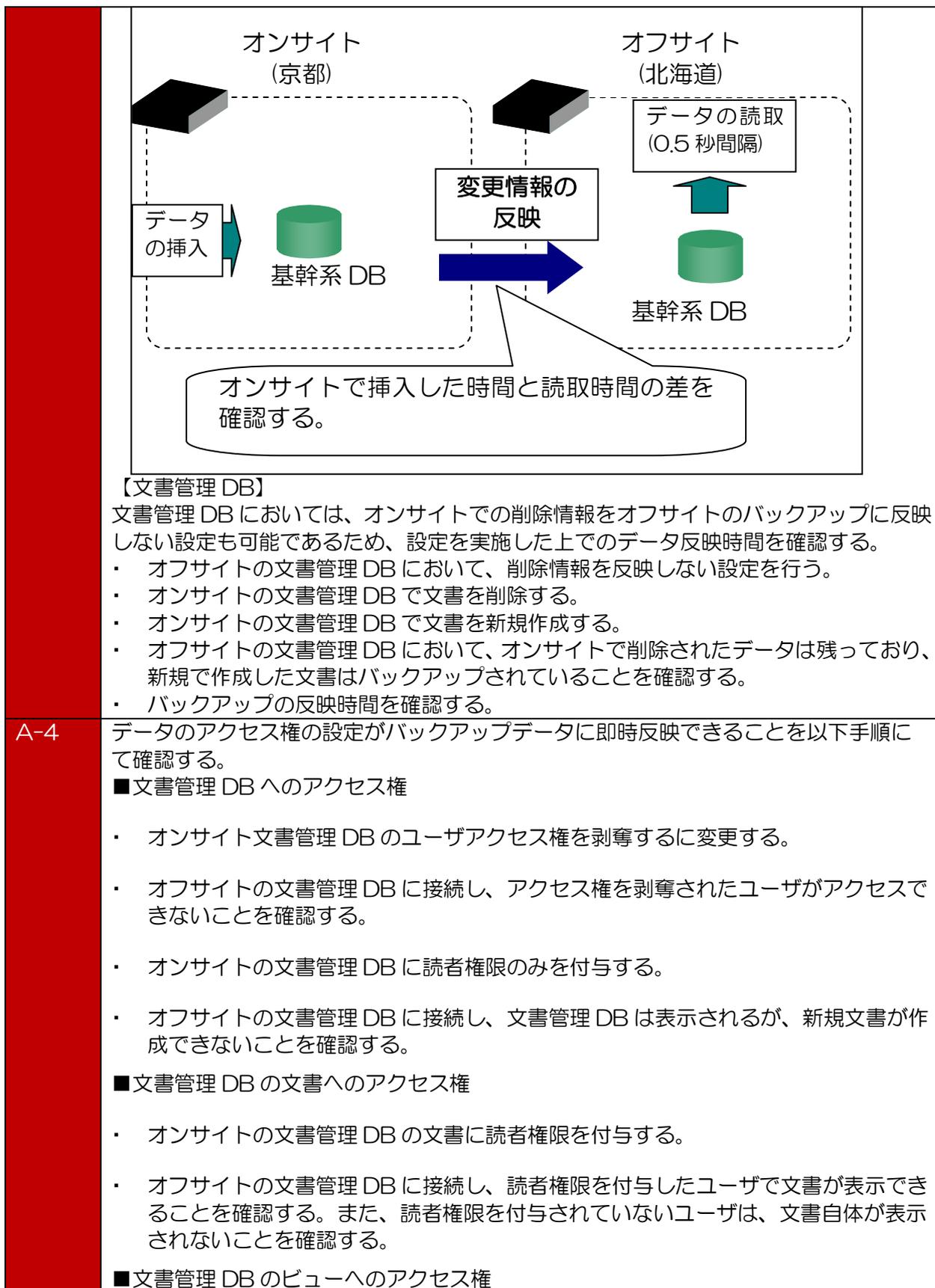
【施策推進支援システム】

- ・ オンサイトの施策推進支援システムの文書管理 DB に、文書を登録する。
- ・ オフサイトの施策推進支援システムの文書管理 DB と基幹系 DB に、データが登録されていることをログと内容により確認する。
- ・ オンサイトの施策推進支援システムの文書管理 DB から文書を削除する。
- ・ オフサイトの施策推進支援システムの文書管理 DB と基幹系 DB から、データが削除されていることをログと内容により確認する。



【基幹系 DB】

大量のデータを一括で挿入することも可能である。今回は、オンサイト側基幹系 DB での 1KB 程度のデータ挿入を 1 秒間隔で、100 回実施し、同期モードの違いによるオフサイト・バックアップでの反映時間を確認する。



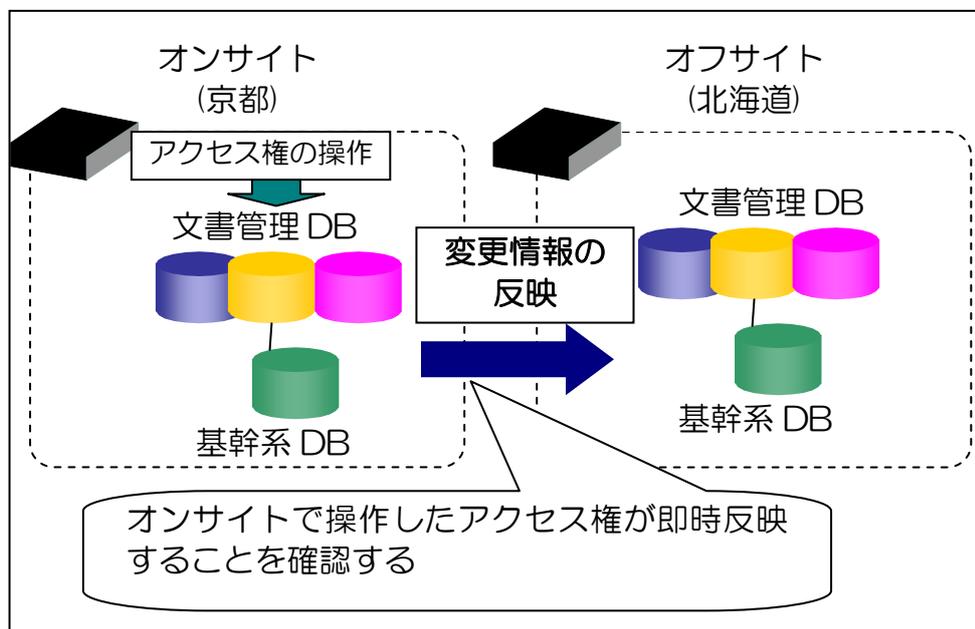
- ・ オンサイトの文書管理 DB のビューに読者権限を付与する。
- ・ オフサイトの文書管理 DB に接続し、読者権限を付与したユーザでビューが表示できることを確認する。また、読者権限を付与されていないユーザは、ビュー自体が表示されないことを確認する。

■文書管理 DB のフィールドへのアクセス権

- ・ オンサイトの文書管理 DB のフィールドに編集権限を付与する。
- ・ オフサイトの文書管理 DB に接続し、編集権限を付与したユーザでフィールドが編集できることを確認する。また、編集権限を付与されていないユーザは、閲覧のみとなることを確認する。

■業務データを利用した状態でのアクセス権変更

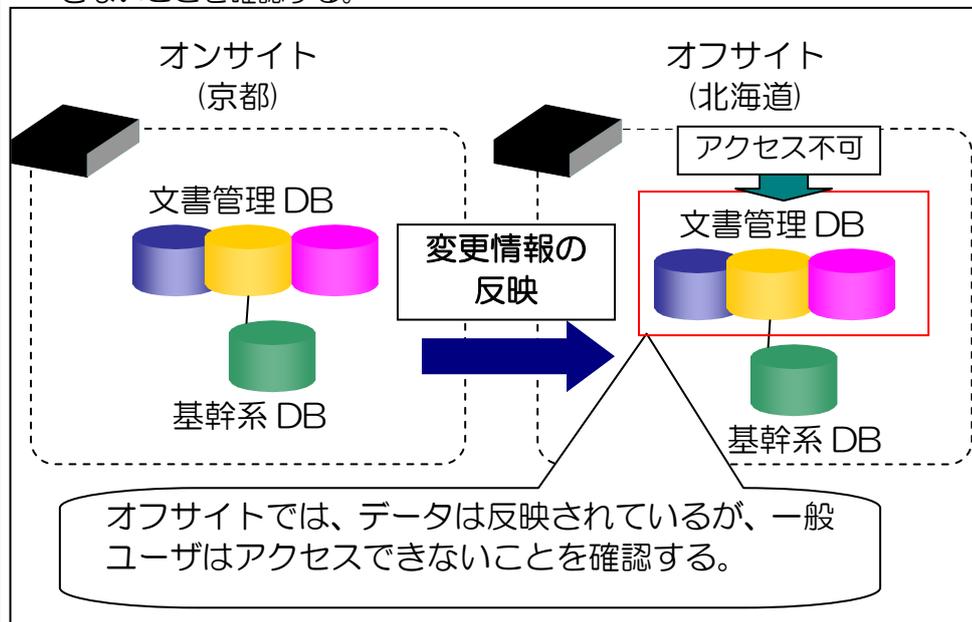
- ・ オンサイトの文書管理 DB で、文書を開いているユーザがいる状態で、その文書のアクセス権を変更する。
- ・ オフサイトの文書管理 DB で、アクセス権が反映されていないことを確認する。
- ・ 5 秒後、アクセスしていたユーザで、文書を閉じる。
- ・ オフサイトの文書管理 DB で、アクセス権が反映されていることを確認する。



また、オフサイトのデータのアクセス権のみを剥奪し、一般ユーザがアクセスできないようにして運用できることを以下手順にて確認する。

- ・ オフサイトの文書管理 DB の 1 つをクラスタ対象外とする。
- ・ オフサイトの複製設定にて、アクセス権は複製しない設定に変更する。
- ・ オフサイトのユーザのアクセス権を剥奪する。
- ・ オンサイトのデータを更新後、複製を実行する。

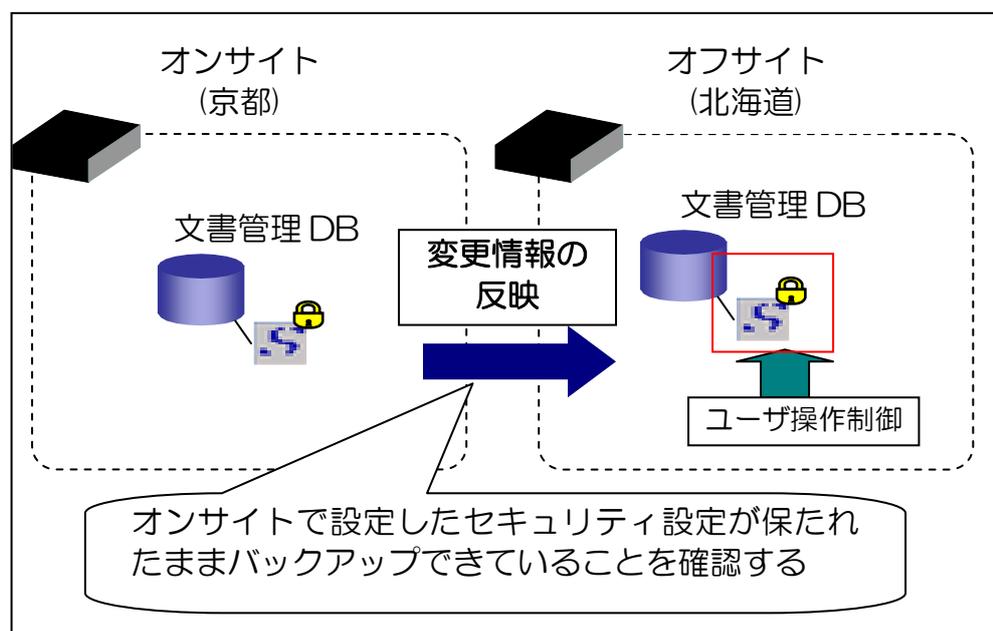
- ・ オフサイトのバックアップに接続しデータが更新されていることを確認する。
- ・ オンサイトにアクセスできるユーザが、オフサイトのバックアップにはアクセスできないことを確認する。



A-5

オフサイト上のデータ形式について、(オンサイトと同様に) 各種機能が有効であることを以下手順にて確認する。

- ・ オンサイトにて、セキュリティ・パッケージを作成し、文書管理 DB に添付する。
- ・ オフサイトのバックアップに、セキュリティ・パッケージが反映されていることを確認する。
- ・ 以下項目に対し、オンサイトで設定した情報のままオフサイトで利用できるか確認する。
  - セキュリティ・パッケージ展開時の (ID, 有効期限による) アクセス・コントロール
  - MS Office などアプリケーションによる操作制限
  - エクスプローラやコマンドプロンプトによるファイル操作制限
  - メールソフトへの添付制限
  - 画面コピー制限
  - 対象外アプリでの操作制限



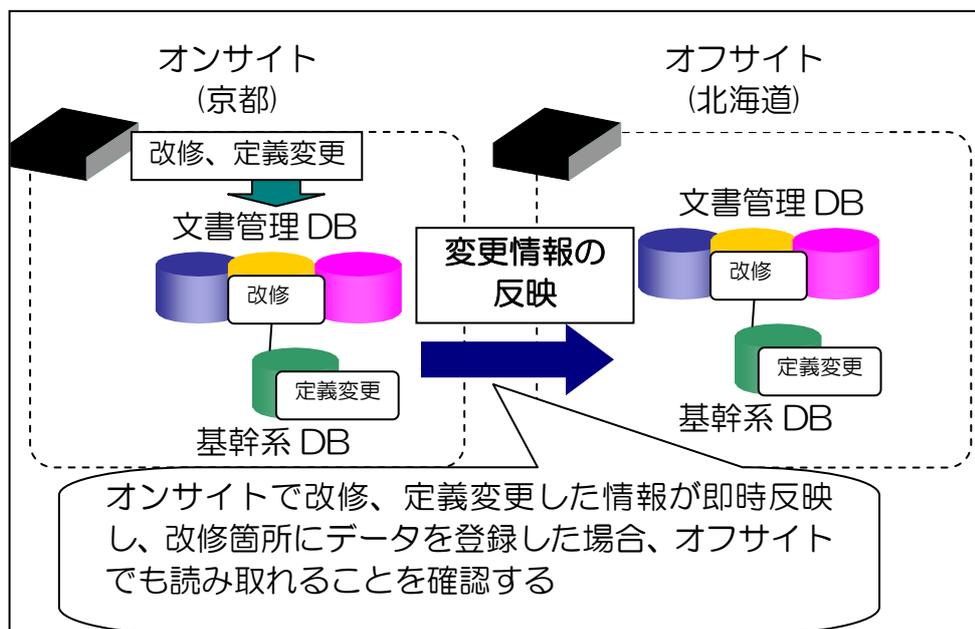
A-6

データベースの設計変更、定義変更が反映されることを以下手順にて確認する。  
【文書管理 DB】

- ・ オンサイトの文書管理 DB にて設計変更(フィールドを追加)を行う。
- ・ 追加したフィールドを利用した文書を作成する。
- ・ オフサイトのバックアップに接続し、新しいフィールドが追加された文書がバックアップされていることを確認する。

【基幹系 DB】

- ・ オンサイトの基幹系 DB にて定義変更(列を追加)を行う。
- ・ 追加した列にデータを挿入する。
- ・ オフサイトのバックアップに接続し、新しい列にデータが追加されていることを確認する。

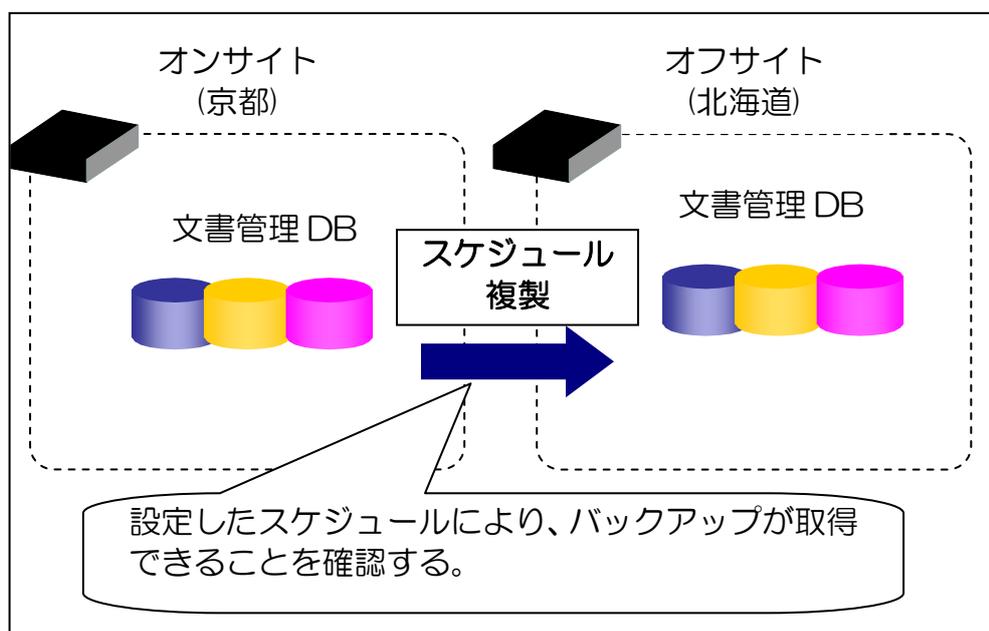


**B** オフサイトへのデータバックアップ運用に関する検証

B-1

夜間バックアップやネットワーク負荷軽減を想定し、スケジュールによるバックアップを以下手順にて確認する。

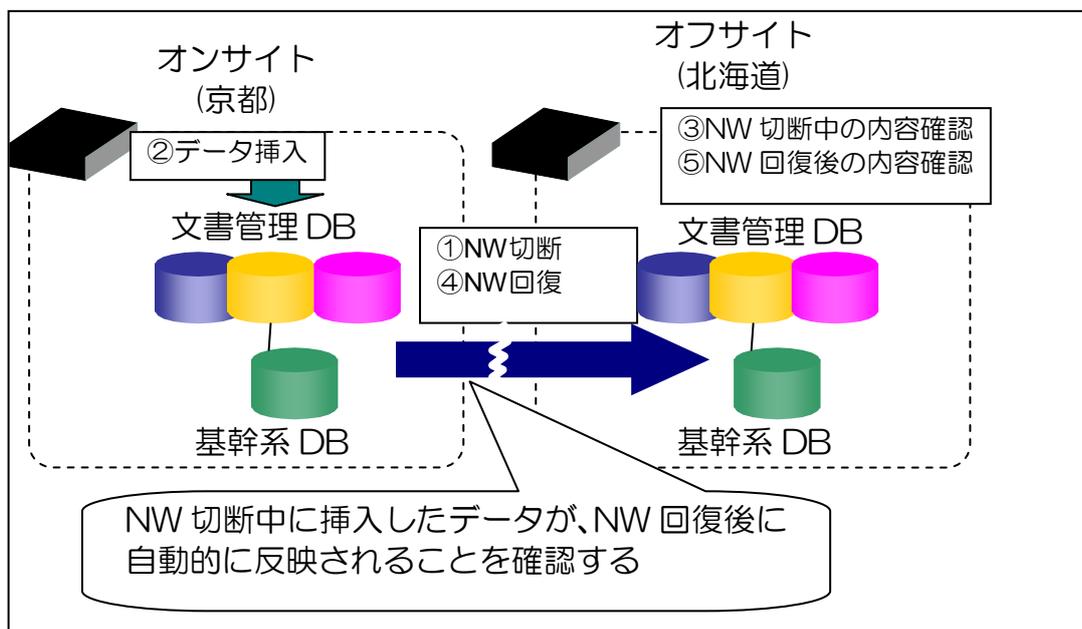
- ・ クラスタ複製を無効にする。
- ・ オンサイトの文書管理 DB にて、10 分に一度複製するというスケジュールを作成する。
- ・ オンサイトの文書管理 DB に新規文書を作成する。
- ・ オフサイトのバックアップは更新されていないことを確認する。
- ・ 10 分経過後、オフサイトのバックアップが更新されていることを確認する。
- ・ 設定した時間に、バックアップが更新されたことをログで確認する。



B-2

通信障害時にバックアップが自動復帰することを以下手順にて確認する。

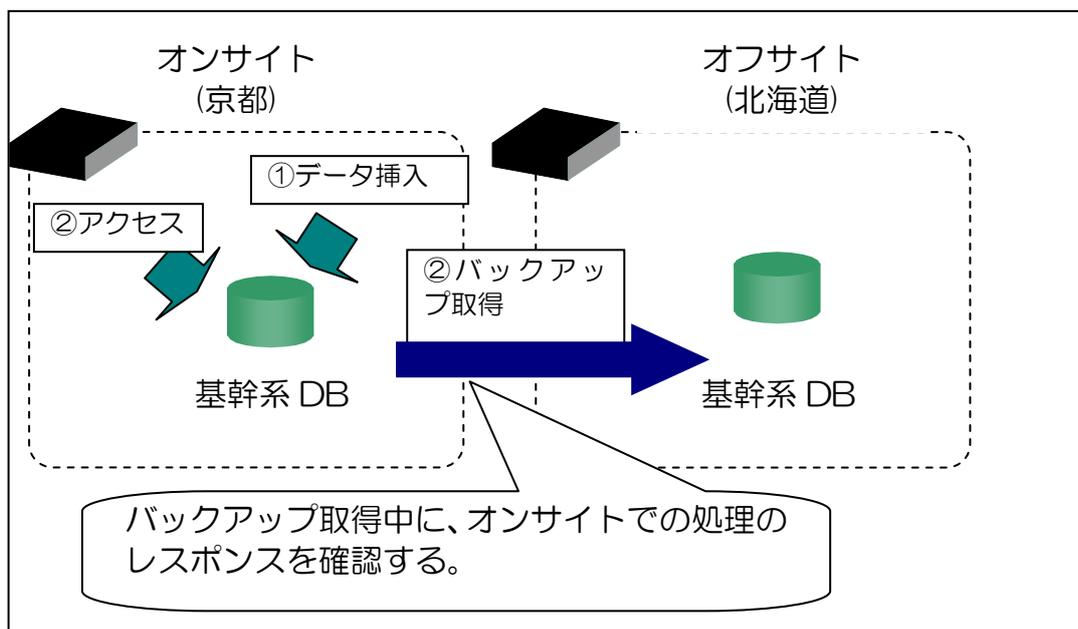
- ・ オンサイトとオフサイト間のネットワークを切断する。
- ・ 文書管理 DB と基幹系 DB にデータを追加する。
- ・ オフサイトのバックアップには追加したデータが反映できていないことを確認する。
- ・ ネットワークを回復させる。
- ・ 回復後、オフサイトのバックアップに接続し、オンサイトで挿入した新規データが挿入されていることを確認する。



B-3

基幹系 DB は大量データの一括データ挿入、更新、照会、削除が運用で実施されることがあるため、同期モードによるオフサイト・バックアップ取得による、オンサイト処理のレスポンス・タイムの影響を確認する。

- ・ HADR によりリアルタイム・バックアップができる状態を構築する。
- ・ オンサイトに、ツールを利用して 1KB 程度のデータ 5000 件を一括で挿入する。
- ・ 挿入中に、オンサイトデータへアクセスし、レスポンス・タイムを取得する。
- ・ 同様に 5000 件のデータを一括で、削除、変更、照会を実施しそれぞれレスポンス・タイムを取得する。
- ・ 上記手順を、同期モードを変更した場合と HADR を利用しない(バックアップを取得しない状態)においてレスポンス・タイムを取得する。



**C オフサイト、オンサイトからのバックアップデータのリストア**

C-1

オフサイト・バックアップからのリストアを以下手順にて確認する。

【文書管理 DB におけるサーバ稼働中のリストア】

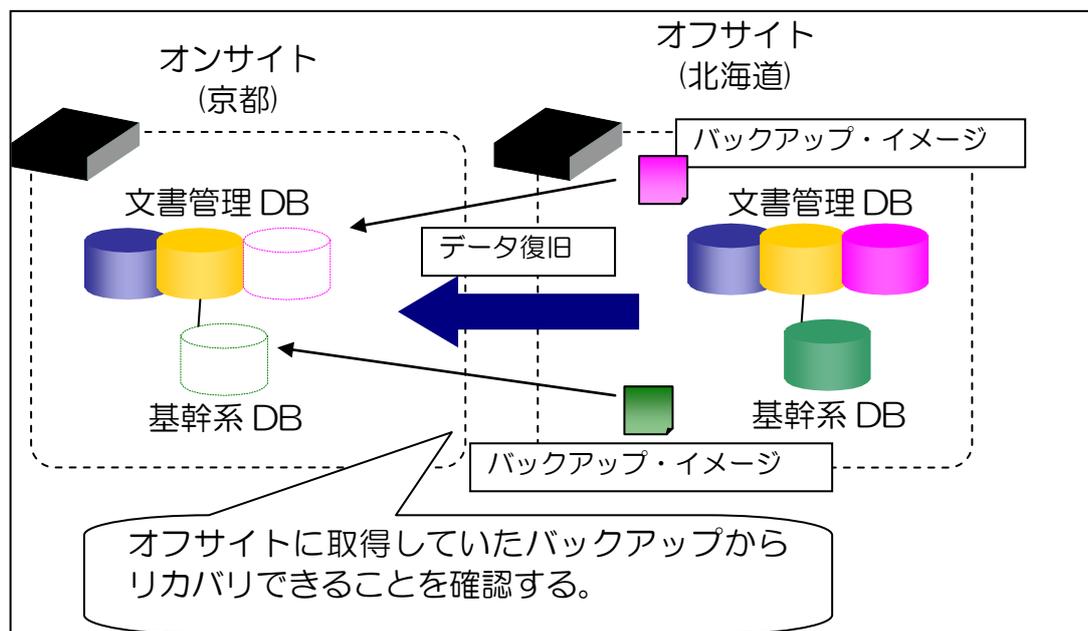
- ・ オンサイトの DB を削除する。
- ・ 手動で、オフサイトの DB をオンサイトに複製し、オンサイトに DB が複製されたことを確認する。
- ・ オンサイト側の DB にアクセスし、削除直前のデータが利用できることを確認する。

【文書管理 DB におけるサーバ停止中のリストア】

- ・ オンサイトの文書管理 DB を削除する。
- ・ オフサイトに保管していたバックアップデータを OS コピーにてオンサイトにコピーする。
- ・ オンサイトから OS コピーしたデータへアクセスし、利用できることを確認する。

【基幹系 DB のオフサイトからのリストア】

- ・ オンサイトの基幹系 DB を削除する。
- ・ オフサイトのバックアップデータを用いてオンサイトに DB 再構築。
- ・ オンサイトとオフサイトの HADR 再開。
- ・ オンサイトのデータへ接続し、利用できることを確認する。



**D オンサイトからのバックアップ・リストア**

D-1

業務アプリケーションのオンサイト・バックアップにおいては、オフサイト・バックアップで確認したサーバ構成の北海道側のサーバを、オンサイトに配置し、2台以上の複数構成とする。その中の1台をユーザからのアクセスを制御したバックアップサーバとして構築することが、一般的であり、実績も多数存在する構成である。以上を踏まえ、オンサイトへのバックアップおよびオンサイトからのリストアについては、以下を検証対象とした。

**【文書管理 DB のオンサイトへのバックアップ】**

- ・ オンサイトのサーバ上に 4MB の文書管理 DB を作成する。
- ・ 手動で複製の設定を実施する。
- ・ 複製を実行する。
- ・ ログと新規文書管理 DB が作成できたことにより、オンサイトにバックアップが取得できたことを確認する。

**【基幹系 DB のオンサイトへのバックアップ】**

- ・ オンサイトのサーバ上に 156MB の基幹系 DB を作成する。
- ・ コマンドを実行し、バックアップを取得する。
- ・ コマンド実行結果と、新規基幹系 DB が作成されていることよりオンサイトにバックアップが取得できたことを確認する。

**【施策推進支援システムのリストア】**

A	<p>上記で取得したオンサイト・バックアップを用い、施策推進支援システムとしてのリストアを以下手順により検証した。</p> <ul style="list-style-type: none"> <li>・ オンサイトに、文書管理 DB と基幹系 DB のバックアップを取得する。</li> <li>・ オンサイトの文書管理 DB と基幹系 DB を削除する。</li> <li>・ オフサイトの文書管理 DB と基幹系 DB それぞれに 1KB のデータを追加する。</li> <li>・ オンサイトに保管しておいたバックアップデータ文書管理 DB をリストアする。</li> <li>・ オンサイトに保管しておいたバックアップから基幹系 DB を再構築する。</li> <li>・ オンサイトのバックアップ時点からの差分をオフサイトから同期を実行する。</li> <li>・ オンサイトのデータへ接続し、利用できることを確認する。</li> </ul>
---	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

ウ) 実証の結果

バックアップ・リストアに関する実証の結果を以下に示す。

A	通常運用時を想定し、自治体クラウドに参加する自治体のデータを、オフサイトにバックアップできることを確認する。																																								
A-1 結果	<p>【文書管理 DB】</p> <p>Lotus Notes/Domino のクラスタ機能を利用することで、事前に作成しておいたオンサイトの 4 つの文書管理 DB(合計 75MB)が 3 分 24 秒で、オフサイトにバックアップが取得できることを確認した。</p> <p>【基幹系 DB】</p> <p>DB2 の HADR サービスを用いることで基幹系 DB(156MB)が 2 分 25 秒で、オフサイトにバックアップが取得できることを確認した。</p> <p>※クラスタ機能、HADR サービスは初期設定にて、随時バックアップを取得する設定となるため、設定を有効にした時点で、オンサイトの DB 作成、更新などのイベント実行時に自動実行される。よって以降の検証についても、手動と明記している箇所以外、コマンドなどを実行する必要はない。</p>																																								
A-2 結果	<p>新規で作成した文書管理 DB(19MB)に対して手動で複製の設定を行い、複製を実行することにより、個別にオフサイトにバックアップが取得できることを確認した。また、オフサイトにアクセスし、事前に登録しておいた文書がオフサイト側のバックアップにも反映されていることを確認した。</p> <p>複製の設定後、複製実行時間に 1 分 35 秒を要した。</p>																																								
A-3 結果	<p>オンサイト側文書管理 DB,基幹系 DB、施策推進支援システムへのデータベースへの以下の操作結果を確認した。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #000080; color: white;"> <th></th> <th style="text-align: center;">操作時間</th> <th style="text-align: center;">データの容量</th> <th style="text-align: center;">反映時間</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">文書管理 DB へのデータ作成</td> <td style="text-align: center;">3 秒</td> <td style="text-align: center;">1KB 未満</td> <td style="text-align: center;">1 秒以内</td> </tr> <tr> <td style="text-align: center;">文書管理 DB へのデータ更新</td> <td style="text-align: center;">2 秒</td> <td style="text-align: center;">1KB 未満</td> <td style="text-align: center;">1 秒以内</td> </tr> <tr> <td style="text-align: center;">文書管理 DB へのデータ削除</td> <td style="text-align: center;">1 秒</td> <td style="text-align: center;">1KB 未満</td> <td style="text-align: center;">1 秒以内</td> </tr> <tr> <td style="text-align: center;">基幹系 DB へのデータ挿入</td> <td style="text-align: center;">2 秒</td> <td style="text-align: center;">1KB 未満</td> <td style="text-align: center;">1 秒以内</td> </tr> <tr> <td style="text-align: center;">基幹系 DB へのデータ更新</td> <td style="text-align: center;">1 秒</td> <td style="text-align: center;">1KB 未満</td> <td style="text-align: center;">1 秒以内</td> </tr> <tr> <td style="text-align: center;">基幹系 DB へのデータ参照</td> <td style="text-align: center;">2 秒</td> <td style="text-align: center;">1KB 未満</td> <td style="text-align: center;">1 秒以内</td> </tr> <tr> <td style="text-align: center;">基幹系 DB へのデータ削除</td> <td style="text-align: center;">1 秒</td> <td style="text-align: center;">1KB 未満</td> <td style="text-align: center;">1 秒以内</td> </tr> <tr> <td style="text-align: center;">施策推進支援システムへのデータ作成(文書管理 DB への反映)</td> <td style="text-align: center;">1 秒</td> <td style="text-align: center;">1KB 未満</td> <td style="text-align: center;">1 秒以内</td> </tr> <tr> <td style="text-align: center;">施策推進支援システムへのデータ作成(基幹系 DB への反映)</td> <td style="text-align: center;">1 秒</td> <td style="text-align: center;">1KB 未満</td> <td style="text-align: center;">1 秒以内</td> </tr> </tbody> </table>		操作時間	データの容量	反映時間	文書管理 DB へのデータ作成	3 秒	1KB 未満	1 秒以内	文書管理 DB へのデータ更新	2 秒	1KB 未満	1 秒以内	文書管理 DB へのデータ削除	1 秒	1KB 未満	1 秒以内	基幹系 DB へのデータ挿入	2 秒	1KB 未満	1 秒以内	基幹系 DB へのデータ更新	1 秒	1KB 未満	1 秒以内	基幹系 DB へのデータ参照	2 秒	1KB 未満	1 秒以内	基幹系 DB へのデータ削除	1 秒	1KB 未満	1 秒以内	施策推進支援システムへのデータ作成(文書管理 DB への反映)	1 秒	1KB 未満	1 秒以内	施策推進支援システムへのデータ作成(基幹系 DB への反映)	1 秒	1KB 未満	1 秒以内
	操作時間	データの容量	反映時間																																						
文書管理 DB へのデータ作成	3 秒	1KB 未満	1 秒以内																																						
文書管理 DB へのデータ更新	2 秒	1KB 未満	1 秒以内																																						
文書管理 DB へのデータ削除	1 秒	1KB 未満	1 秒以内																																						
基幹系 DB へのデータ挿入	2 秒	1KB 未満	1 秒以内																																						
基幹系 DB へのデータ更新	1 秒	1KB 未満	1 秒以内																																						
基幹系 DB へのデータ参照	2 秒	1KB 未満	1 秒以内																																						
基幹系 DB へのデータ削除	1 秒	1KB 未満	1 秒以内																																						
施策推進支援システムへのデータ作成(文書管理 DB への反映)	1 秒	1KB 未満	1 秒以内																																						
施策推進支援システムへのデータ作成(基幹系 DB への反映)	1 秒	1KB 未満	1 秒以内																																						

施策推進支援システムへのデータ削除(文書管理 DB への反映)	1 秒	1KB 未満	1 秒以内
施策推進支援システムへのデータ削除(基幹系 DB への反映)	1 秒	1KB 未満	1 秒以内

また、文書管理 DB へのファイルアップロードに関しては、ファイル容量が大きい場合、ファイルの容量およびネットワーク性能に応じて、オフサイトの反映に一定の時間がかかることを確認した。

ファイルサイズ	反映時間
1 MB	5 秒
100MB	10 分 12 秒
1GB	1 時間 51 分 39 秒

【基幹系 DB】

基幹系 DB での大量データ更新時におけるバックアップへの更新伝播時間の同期モードによる違いは次の通りだった。大量データ更新時として、1KB 未満のデータを 1 秒間隔で 100 回挿入し、同期モードとしての違いを確認した。ただし、0.5 秒単位でモニターしたため、±0.5 秒の誤差を含む。

同期モード	同期	近似同期	非同期
オンサイトとオフサイトの差異	1.223912	2.87751	2.3428844

(表中の数字の単位は秒)

【文書管理 DB】

削除情報を反映しない設定を行い、オフサイトのバックアップへの反映状況を以下手順にて確認した。設定や文書削除などの処理は手動で行い、2,3 秒で完了する程度の操作である。

- ・ オフサイトの文書管理 DB において、削除情報を反映しない設定を行う。
- ・ オンサイトの文書管理 DB で文書を削除する。
- ・ オンサイトの文書管理 DB で文書を新規作成する。
- ・ オフサイトの文書管理 DB で、新規作成した文書と、オンサイトでは削除した文書の両方が残っていることを確認した。

また新規で登録した文書は、1KB 未満であり、バックアップへの反映は 1 秒以内に完了することが確認できた。

A-4 結果

オンサイトの文書管理 DB へのアクセス制御設定変更が、オフサイトバックアップ反映されることを以下により確認した。アクセス制御設定変更操作は 3 秒程度で完了する。アクセス権の設定変更を実施したユーザを対象ユーザ、その他ユーザを一般ユーザとする。

操作	複製前		複製後	
	対象ユーザ	一般ユーザ	対象ユーザ	一般ユーザ
オンサイトの文書管理 DB のアクセス権剥奪	アクセス可能	アクセス可能	アクセス不可	アクセス可能
オンサイトの文書管理 DB に読者権限を付与	アクセス不可	編集まで可能	閲覧のみ可能	編集まで可能
オンサイトの文書管理 DB の文書に読者権限にて制限	対象文書が表示	対象文書が表示	対象文書の閲覧が可能	対象文書が非表示
オンサイトの文書管理 DB のビューを読者権限にて制限	対象ビューが表示	対象ビューが表示	対象ビューの閲覧が可能	対象ビューが非表示
オンサイトの文書管理 DB のフィールドを読者権限にて制限	対象フィールドの編集可能	対象フィールドの編集が可能	対象フィールドの閲覧が可能	対象フィールドが非表示

上記より、権限がないユーザは、対象の文書やビューが非表示となり、閲覧ができない

結果となった。

また、バックアップへの設定変更反映時間を以下に示す。オンサイトの文書管理にアクセスした状態で、アクセス権を変更した場合は、アクセスしているユーザが画面遷移したタイミングで、バックアップへ反映されるため、多少時間がかかる場合があることが確認できた。“オンサイトの文書管理 DB に別のユーザでアクセスしたまま、文書に読者権限付与”については、5 秒後に、表示していた文書を閉じたため、反映時間が 5 秒となった。

操作	反映時間
オンサイトの文書管理 DB のアクセス権剥奪	1 秒以内
オンサイトの文書管理 DB に読者権限を付与	1 秒以内
オンサイトの文書管理 DB の文書に読者権限にて制限	1 秒以内
オンサイトの文書管理 DB のビューを読者権限にて制限	1 秒以内
オンサイトの文書管理 DB のフィールドを読者権限にて制限	1 秒以内
オンサイトの文書管理 DB に別のユーザでアクセスしたまま、文書を読者権限にて制限	5 秒

オフサイトのみアクセス権剥奪については、オフサイトの設定を変更することで、オフサイトの文書管理 DB にはアクセスできないことを確認した。ただし複製を実行すると、データは更新されていることを確認した。

A-5 結果

下記セキュリティ設定において、オンサイトに設定した情報のまま、正常にバックアップが取得されることが確認できた。

操作制限	オンサイト	オフサイト
設定したユーザ ID/パスワードによる展開	展開可能	展開可能
設定していないユーザ ID/パスワードによる展開	展開不可	展開不可
有効期限内の展開	展開可能	展開不可
有効期限外の展開	展開不可	展開不可
MS Office 等による上書き保存の制限	上書き不可	上書き不可
MS Office 等による名前を付けて保存の制限	保存不可	保存不可
MS Office 等によるコピー&ペーストの制限	コピー&ペースト不可	コピー&ペースト不可
印刷の制限	印刷不可	印刷不可
エクスプローラによるファイル複写制限	複写不可	複写不可
エクスプローラによるファイル移動制限	移動不可	移動不可
Notes クライアントのメールへの添付制限	添付不可	添付不可
Outlook Express のメールへの添付制限	添付不可	添付不可
画面コピーの制御	操作不可	操作不可
対象外アプリケーションでの表示制限	表示不可	表示不可

A-6 結果

【文書管理 DB】  
設計変更がバックアップにも反映されることを、以下手順にて確認した。

- ・ オンサイトの文書管理 DB でフィールドを追加
- ・ 1KB 未満の文書を新規作成し、追加したフィールドにデータを登録し、保存

	<ul style="list-style-type: none"> <li>・ オフサイトの文書管理 DB にアクセスし、新規の文書が登録されていることを確認</li> <li>・ 設計を確認し、オフサイトの文書管理 DB にもフィールドが追加されていることを確認</li> </ul> <p>また、オンサイトでの文書作成時間より、1 秒以内に、オフサイトの文書管理 DB に新規文書が登録されていることを確認できた。</p> <p>【基幹系 DB】 定義変更がバックアップにも反映されることを以下手順にて確認した。</p> <ul style="list-style-type: none"> <li>・ オンサイトの基幹系 DB にて定義変更(列を追加)を行う</li> <li>・ 追加した列にデータを挿入する</li> <li>・ オフサイトのバックアップにアクセスし、新しい列が追加されていることを確認</li> <li>・ 追加した列にデータが登録されていることを確認</li> </ul> <p>また、オンサイトの基幹系 DB にデータ追加時間より、1.5 秒以内に、オフサイトの基幹系 DB にデータが登録されたことが確認できた。</p>						
<b>B</b>	<b>オフサイトへのデータバックアップ運用に関する検証</b>						
B-1 結果	<p>文書管理 DB において、設定したスケジュールに従い、10 分毎にオフサイトにバックアップが取得できることを確認した。1 度データを作成した後は、10 分の間にデータの追加はないものとした。</p> <table border="1" data-bbox="352 1240 1233 1352"> <thead> <tr> <th>操作</th> <th>反映時間</th> </tr> </thead> <tbody> <tr> <td>リアルタイム同期時のデータ作成(1KB 未満)</td> <td>1 秒以内</td> </tr> <tr> <td>スケジュール同期時のデータ作成(1KB 未満)</td> <td>1 秒以内</td> </tr> </tbody> </table>	操作	反映時間	リアルタイム同期時のデータ作成(1KB 未満)	1 秒以内	スケジュール同期時のデータ作成(1KB 未満)	1 秒以内
操作	反映時間						
リアルタイム同期時のデータ作成(1KB 未満)	1 秒以内						
スケジュール同期時のデータ作成(1KB 未満)	1 秒以内						
B-2 結果	<p>通信障害復旧後、自動で回復できることを以下にて確認した。</p> <ul style="list-style-type: none"> <li>・ 通信障害を発生させるため、文書管理 DB、基幹系 DB とともに LAN ケーブルを抜くことにより、一時的なネットワークの切断状況とした。</li> <li>・ オンサイトの文書管理 DB のデータと基幹系 DB のデータを更新</li> <li>・ オフサイトのデータには反映されていないことを確認できた。</li> <li>・ LAN ケーブルを接続し、オフサイトのデータを確認。</li> </ul> <p>ネットワーク回復後、1 秒以内に、文書管理 DB、基幹系 DB とともにデータは自動更新されることを確認した。</p>						
B-3 結果	<p>HADR サービスの同期モードを変更し、1KB 程度のデータを 5000 件 挿入、更新、照会、削除した際のレスポンスタイムは下表の通りだった。</p>						

同期モード		HADR なし	同期	近似同期	非同期
Insert (挿入)	平均	0.004649	0.066652	0.063046	0.030981
	HADR なしとの差	—	0.062004	0.058397	0.026332
Update (更新)	平均	0.004162	0.063088	0.058983	0.028980
	HADR なしとの差	—	0.058926	0.054821	0.024818
Select (参照)	平均	0.000155	0.000161	0.000160	0.000160
	HADR なしとの差	—	0.000005	0.000004	0.000004
Delete (削除)	平均	0.002544	0.036922	0.034512	0.016640
	HADR なしとの差	—	0.034378	0.031969	0.014096

(表中の数字の単位は秒)

**C オフサイトからのバックアップデータのリストア**

**C-1 結果** 【文書管理 DB】  
 オフサイトバックアップからの、サーバ稼動中におけるリストアとして以下手順で確認した。

- ・ オンサイトの 4MB の文書管理 DB を削除
- ・ オフサイトのバックアップから複製を作成
- ・ 複製を実行
- ・ オンサイトの文書管理 DB が利用できることを確認

オフサイトバックアップからの、サーバ停止中におけるリストアとして以下手順で確認した。

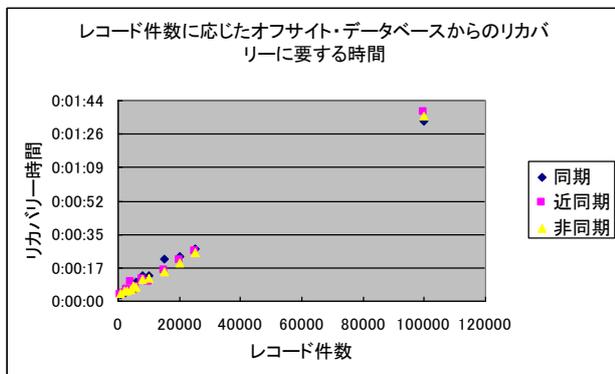
- ・ オンサイトの 4MB の文書管理 DB を削除
- ・ オフサイトのバックアップを OS コピーにて、オンサイトに配置
- ・ オンサイトの文書管理 DB が利用できることを確認

容量	操作	リストア時間
4MB の文書管理 DB	オフサイトのバックアップを利用し、複製実行にて復旧	20 秒
4MB の文書管理 DB	オフサイトのバックアップを OS コピーにてオンサイトで復旧	3 秒

【基幹系 DB】  
 オフサイトのデータを利用し、オンサイトに DB を再構築する方法を検証し、100MB の DB が、10 分で復旧できたことを確認した。その他レコード件数(データ量)に応じ、リカバリに要する時間は以下の通りである。

同期モード		同期	近似同期	非同期
レコード件数 (生データ量)	1000 (約 50KB)	0:00:03	0:00:03	0:00:04
	2000 (約 100KB)	0:00:04	0:00:04	0:00:05
	3000 (約 150KB)	0:00:06	0:00:06	0:00:05

4000 (約 200KB)	0:00:06	0:00:10	0:00:06
5000 (約 250KB)	0:00:07	0:00:05	0:00:08
6000 (約 300KB)	0:00:10	0:00:07	0:00:07
8000 (約 400KB)	0:00:13	0:00:11	0:00:11
10000 (約 500KB)	0:00:13	0:00:10	0:00:12
15000 (約 750KB)	0:00:22	0:00:16	0:00:15
20000 (約 1MB)	0:00:23	0:00:21	0:00:20
25000 (約 1.25MB)	0:00:27	0:00:26	0:00:25
100000 (約 5MB)	0:01:33	0:01:38	0:01:36



D-1 結果

【文書管理 DB のオンサイトバックアップ】

- 新規で作成した文書管理 DB(4MB)に対して手動で複製の設定を行い、複製を実行することにより、個別にオンサイトにバックアップが取得できることを確認した。また、オンサイトのバックアップにアクセスし、事前に登録しておいた文書が、オンサイト側のバックアップにも反映されていることを確認した。複製の設定後、複製実行時間に 9 秒を要した。

【基幹系 DB のオンサイトバックアップ】

- 新規で作成した基幹系 DB(156MB)に対してバックアップのコマンドを実行し、オンサイトにバックアップが取得できることを確認した。またオンサイトのバックアップにアクセスし、事前に登録しておいた文書が、オンサイト側のバックアップにも反映されていることを確認した。バックアップ実行時間は 20 秒を要した。

【施策推進支援システムのリストア】

- リカバリ時間の短縮のため、文書管理 DB および基幹系 DB で定期的を取得するオンサイトバックアップを用い、リストアする。その後、オンサイトでのバックアップ時点と障害発生までの差分をオフサイトからの同期により、整合性をとる方法を検証した。オンサイトにおける 4MB の文書管理 DB のリストアについては、サー

バ稼動中の場合、バックアップ時と同様の9秒を要した。下記表での4MBの文書管理DBのリストアはサーバ停止中のリストア時間を記述する。

リストア手順	所要時間
オンサイトの4MBの文書管理DBをオンサイトにバックアップ	9秒
オンサイトの156MBの基幹系DBをオンサイトにバックアップ	20秒
オンサイトの文書管理DBと基幹系DBを削除	3秒
オフサイトの文書管理DBと基幹系DBにそれぞれ1KBのデータを挿入	3秒
オンサイトに取得していたバックアップからの4MBの文書管理DBリストア	1秒
オンサイトに取得していたバックアップからの100MBの基幹系DBリストア	3分40秒
オンサイトとオフサイトの情報の同期	2秒
リストア手段による比較	所要時間
オフサイトからの4MBの文書管理DB複製作成+100MBの基幹系DBの再構築	10分20秒
オンサイトからの4MBの文書管理DB複製作成+100MBの基幹系DBの再構築+同期	3分43秒

## エ) 実証の考察

### A. バックアップ機能について

バックアップ手段については、以下結果より、複数のバックアップ手段から参加自治体のバックアップ要件に合致する方法を選択でき、組み合わせることにより、大量データを扱う業務アプリケーションは、スケジュールで夜間にのみバックアップ、リアルタイムの同期性を求められるDBは、クラスターやHADRを用いたバックアップ、など参加自治体のバックアップ要件が異なる場合にも有効な手段となる。

また、文書管理DBは、75MBのオフサイトバックアップに3分24秒かかっているが、基幹系DBは、156MBでも2分25秒でバックアップができており、基幹系DBの方が、バックアップ速度が速いことがわかる。これは製品の仕様における差異であり、文書管理DBは、1つの棚に、複数の種類の物を詰め込むように、フィールドを後から追加することや、1つのフィールドに対して、ファイルや文字列を混在させることが可能である。情報共有として、ユーザがどのようなデータを追加したいかが明確でない場合、複数の情報を管理したい場合において優れたDBである。ただし、整理されていないため、検索には不向きである。対して、基幹系DBは、1つの棚には1つの種類の物を格納するように、フィールドを定義し、必要な情報を管理するDBである。フィールドを定義することで、このフィールドには、何文字の文字列が入っているなどが把握できているため、検索時のキーの絞り込みができ、大容量データ参照や検索に優れたDBとなっている。管理したいデータに文字列や数字が多く、データの参照、更新の即時性を重視する場合には、基幹系DBを利用した方がオンサイトとオフサイトのデータ同期率が高く、障害直前の情報までオフサイトにバックアップできる。

表 4-48 バックアップ手段と所要時間

バックアップ手段	対象 DB	所要時間	検証番号
一括バックアップ	4つの文書管理 DB(計 75MB)+基幹系 DB(156MB)	5分49秒	A-1
	4つの文書管理 DB(計 75MB)	3分24秒	A-1
個別バックアップ(設定により自動実行もしくは手動での適宜実行が可能)	文書管理 DB(19MB)	1分35秒	A-2
	基幹系 DB(156MB)	2分25秒	A-1

またオンサイトバックアップとオフサイトバックアップを比較すると、オンサイトの方が早いことがわかる。これはネットワーク状況に依存していると言えるため、バックアップしたい対象や重要性、災害時の業務継続を検討し、どちらにバックアップを取得するか、または両方にバックアップする運用とするかなどの検討が必要となる。

表 4-49 バックアップ手段と所要時間

バックアップ手段	対象 DB	所要時間	検証番号
オンサイトバックアップ	文書管理 DB(4MB)	9秒	G-1
	基幹系 DB(156MB)	30秒	C-1
オフサイトバックアップ	文書管理 DB(19MB)	1分35秒	A-2
	基幹系 DB(156MB)	2分25秒	A-1

## B. データ反映の即時性と正確性について

一度オフサイトにバックアップを取得すれば、その後のデータの一部更新やアクセス権変更においては、以下結果より、正確に、数秒以内を保ってバックアップへ反映できていると言える。また、文書管理 DB では 1MB のデータ反映に 5 秒かかっているが、基幹系 DB では 100KB(1KB 程度のデータを 100 件)挿入した結果でも 1.2 秒以内にデータ反映ができた。バックアップだけでなく、データ反映の即時性という観点からも大容量データを利用する場合には、基幹系 DB を活用することにより、即時性は高まると言える。ちなみに、基幹系 DB には同期モードも複数から選択でき、近似同期でも 100KB (1KB 程度のデータを 100 件)のデータ更新であれば平均 2.8 秒の時間差でオフサイトのバックアップへデータ反映ができています。業務要件で、数秒の誤差が許容できる場合は、非同期モードや近似同期モードを利用することで、秒単位の更新量が分散され、ネットワーク負荷を軽減することもできる。

下記表中の基幹系 DB については、1KB 程度のデータを 100 件データ更新したため、100KB のデータ更新と記述する。

表 4-50 データの反映時間

操作	オンサイトとオフサイトの 内容比較	反映時間	検証 番号
文書管理 DB でのデータの新規追加	オンサイトと同様	1 秒	A-3
文書管理 DB でのデータの変更	オンサイトと同様	1 秒	A-3
文書管理 DB でのデータの削除	オンサイトと同様	1 秒	A-3
文書管理 DB でのアクセス権	オンサイトと同様	1 秒	A-4
文書管理 DB に添付したセキュリティ・データ(1MB)	オンサイトと同様	5 秒	A-5
文書管理 DB に添付したセキュリティ・データ(100MB)	オンサイトと同様	10 分 12 秒	A-3
基幹系 DB での 100KB のデータ更新 (同期モード)	オンサイトと同様	1.2 秒	A-3
基幹系 DB での 100KB のデータ更新 (近似同期モード)	オンサイトと同様	2.8 秒	A-3

ただし、文書管理 DB の場合、100MB の添付ファイルの反映に、10 分かかったという結果が出ている。大量データのバックアップには注意が必要になる。

#### C. セキュリティについて

A-4 の文書管理 DB 単位や、文書単位、ビュー単位、フィールド単位でのアクセス権設定がバックアップに反映される結果や、A-5 のオンサイトで設定したセキュリティ情報が付与された状態でバックアップが取得できるという結果より、オフサイトのバックアップもセキュリティを強化した状態で保管できていると言える。オフサイトのバックアップデータを守るために、まずはアクセス権によるセキュリティ強化を実施する。そして文書管理 DB から抜き出した後のデータの情報漏えいを防ぐためにセキュリティ・パッケージ化したデータを文書管理 DB に添付する。このような運用を行うことにより、自治体クラウドにおける情報漏えいの防止に役立つ仕組みが構築できる。

#### D. 通信障害対応について

通信障害が発生した場合でもネットワーク復旧後、自動で回復することが B-1 の検証にて実証できた。この仕組みを利用することにより、各自治体で、ネットワーク通信障害が発生した場合においても、自治体クラウド提供者は特に対応をせず、自動でバックアップの仕組みが復旧でき、運用負荷を軽減できる。

#### E. 大量データ更新中におけるオンサイト利用レスポンスについて

基幹系 DB は、自動プログラムによる大量データの自動更新が発生する場

合がある。5000 件のデータ挿入、更新、照会、削除のレスポンスタイムを取得した B-3 の結果より以下の傾向が見られる。

- ・ オフサイト・データベースへのバックアップを行わないケースと比べて、いずれの同期モードでも 0.06 秒程度のレスポンス劣化にとどまる。
- ・ データ変更操作(挿入、更新、削除)については、同期(Sync)と近似同期(NearSync)の差は小さい。
- ・ データ変更操作について、非同期と同期および近似同期の差は大きいですが、それでも 0.04 秒以内の差である。
- ・ データ参照に関しては、同期モードによる差はない。

以上のことより、今回実証を行った環境においても、0.1 秒以内程度のレスポンスタイムが許される業務であれば、どの同期モードを用いてもオフサイト・データベースによるバックアップシステムを利用することができる。

また多少でもレスポンスタイムの劣化が好ましくないような業務においては、同期モードとしての非同期(Async)を用いることで、レスポンスタイムの劣化を 0.03 秒に抑えることができる。また参照が主な業務においては、実質上のレスポンスタイムの劣化は感じ取れないだろうと考えられる。またデータ変更操作については今回の実証は、15 トランザクション/秒程度のトランザクション量に相当する。それ以上のアクセスがあるようなケースではレスポンス劣化も懸念されるため、本番運用時には、どの程度のトランザクションが発生するかによって、DB の構成やハードウェアスペック、ネットワーク帯域など検討する必要がある。

#### F. バックアップデータからのリストアについて

C-1 のオフサイトにあるデータを用いて、リストアが可能であることが確認できた結果と D-1 のオンサイトにあるデータを用いて、リストアが可能であることが確認できた結果より、どちらのバックアップデータを用いても、リストア可能であることがわかる。ただし、オンサイトのバックアップデータを用いた方がリストア時間は短縮可能である。また以下結果より、サーバは停止していた方がよりリストア時間は短くなる。復旧時間を短くするためには、オンサイトにあるバックアップを用い、サーバを停止してリストアすることが有効であることがわかる。障害からの短時間復旧が望まれる業務アプリケーションにおいては、オンサイトのバックアップからリストアし、オフサイトのバックアップとデータの同期をとる手段が有効である。

表 4-51 リストア時の反映時間

リストア手段		反映時間	検証番号
4MB の文書管理 DB をオフサイトからリストア	サーバ停止中	3 秒	C-1
	サーバ稼動中	20 秒	C-1

4MBの文書管理DBをオンサイトからリストア	サーバ停止中	1秒	D-1
	サーバ稼動中	9秒	D-1
100MBの基幹系DBをオンサイトからリストア	サーバ稼動中	3分40秒	D-1
100MBの基幹系DBをオフサイトからリストア	サーバ稼動中	10分	C-1

(5) バックアップ・リストアに関するその他有用な機能の実証

ア) 実証の概要・目的

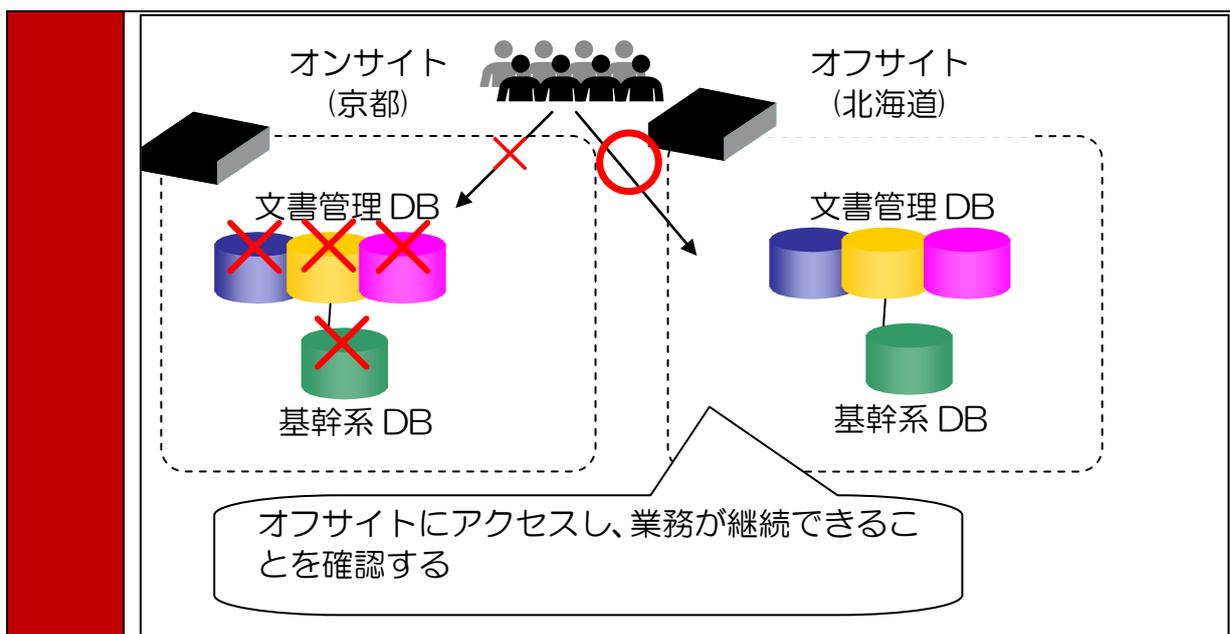
バックアップ・リストア機能に関するその他有効な機能について以下を目的に実証を実施した。

- E) オンサイト災害時のオフサイトでの業務継続の為、オフサイトのデータ利用が可能かどうか、およびレスポンス状況を確認
- F) メンテナンス時においても業務が継続できるようにする為、業務を停止しないメンテナンス方法を確認
- G) 各自治体において、管理者による文書の一括削除時に、誤って文書を削除してもすぐ復旧できるためのバックアップや、設計変更時の影響確認に利用するなどの要件対応の為、管理者のクライアントマシンを用いたバックアップ・リストアを確認

イ) 実証の内容

バックアップ・リストアに関するその他有用な機能の実証内容について以下に示す。

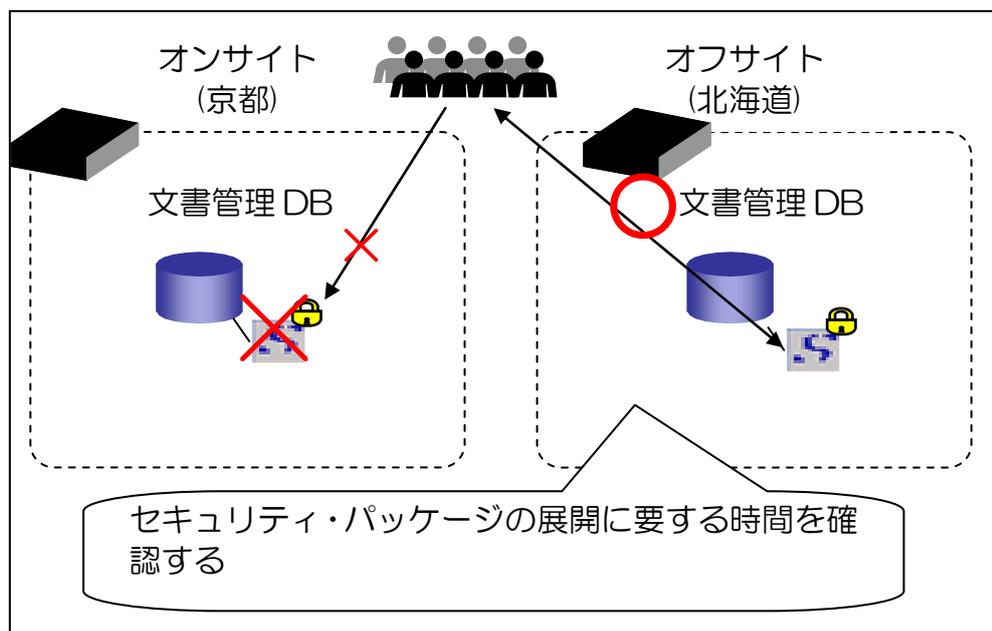
E	<b>災害時オフサイト側システムでの業務継続性</b>
E-1	<p>オンサイト災害時に、オフサイトでの業務継続が可能なことを確認する。</p> <ul style="list-style-type: none"> <li>・ オンサイトの文書管理DBと基幹系DBを削除する。</li> <li>・ オンサイトの災害を想定し、クラスタ機能を停止する。</li> <li>・ HADRサービスのプライマリをオフサイトに切り替える。</li> <li>・ オフサイトのデータにアクセスし、利用していたオンサイトと同様のデータが登録されていることを確認する。</li> <li>・ オフサイトのデータに対して、文書作成を実施し、同じように利用できることを確認する。</li> </ul>



E-2

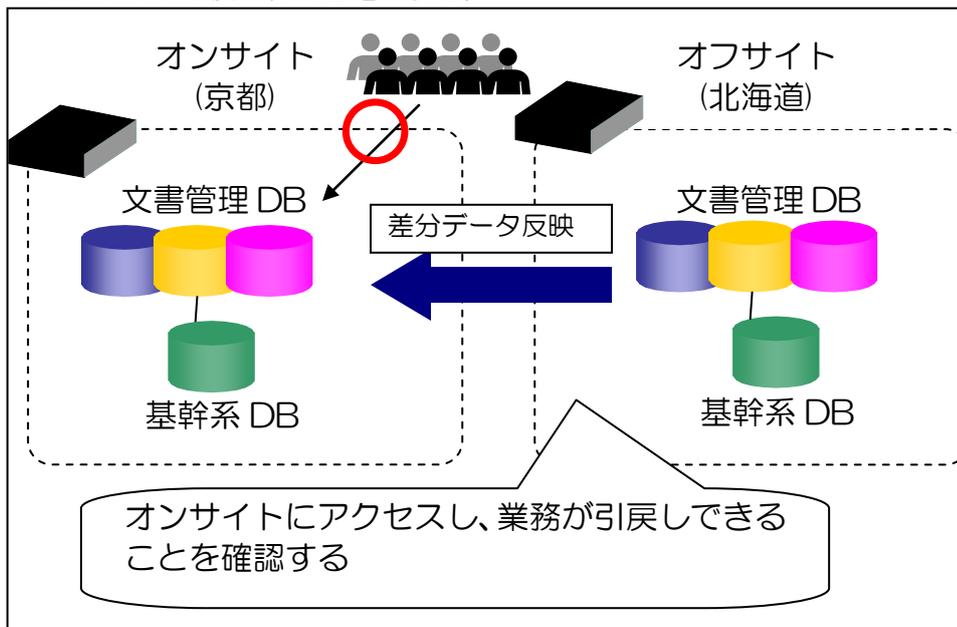
オフサイト上のセキュリティ・パッケージのレスポンスを確認する。

- ・ オンサイトで、10MB のセキュリティ・パッケージの展開時間を計測
- ・ オフサイトにバックアップされている、10MB のセキュリティ・パッケージの展開時間を計測
- ・ 上記について、セキュリティ・パッケージのサイズを 50MB,100MB に変更しそれぞれ展開時間を計測
- ・ オンサイトで 10MB の ZIP 圧縮の展開時間を計測
- ・ オフサイトにバックアップされている ZIP 圧縮の展開時間を計測
- ・ ZIP 圧縮ファイルについても、ファイルサイズを 50MB,100MB に変更し、それぞれ展開時間を計測

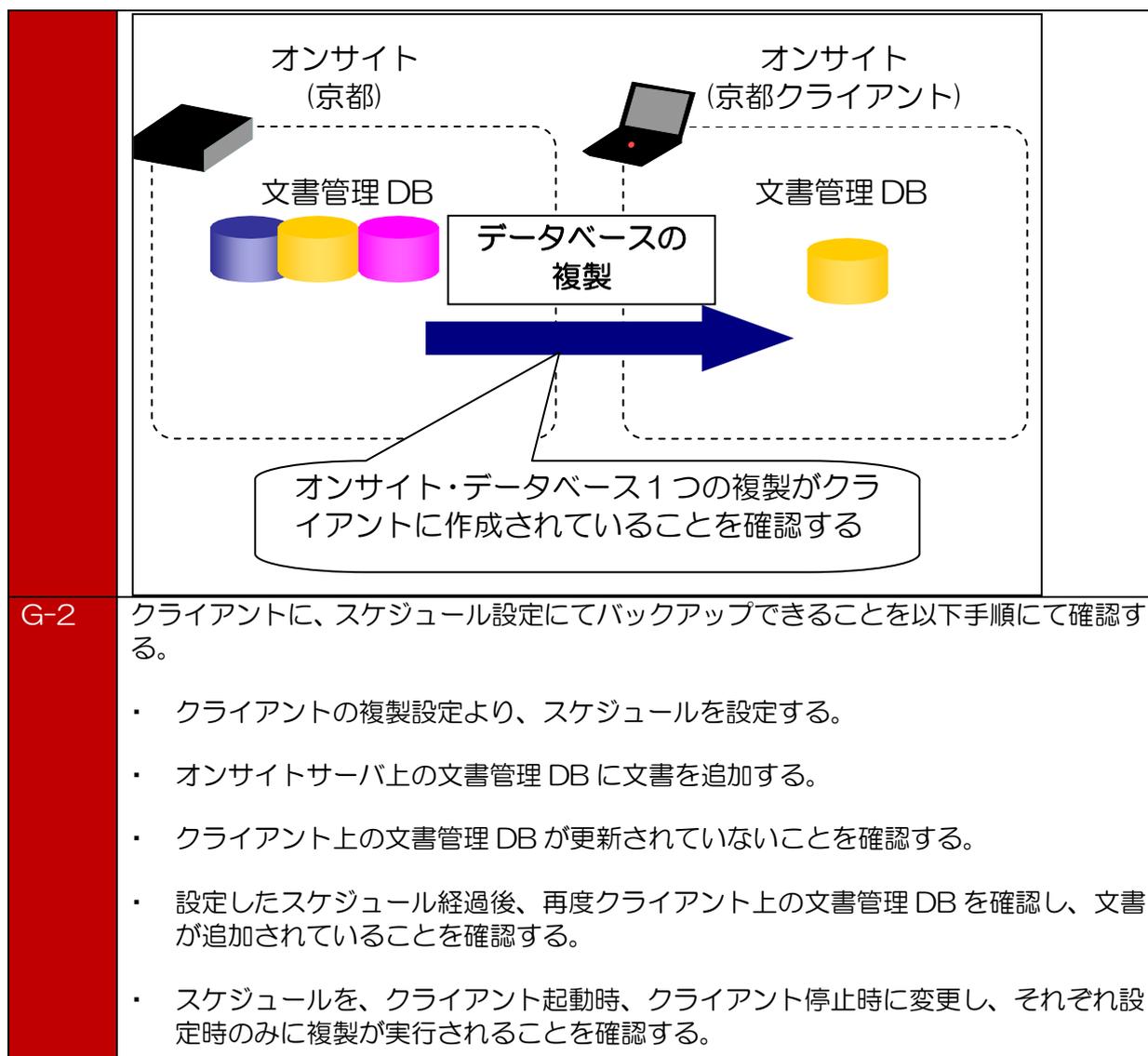


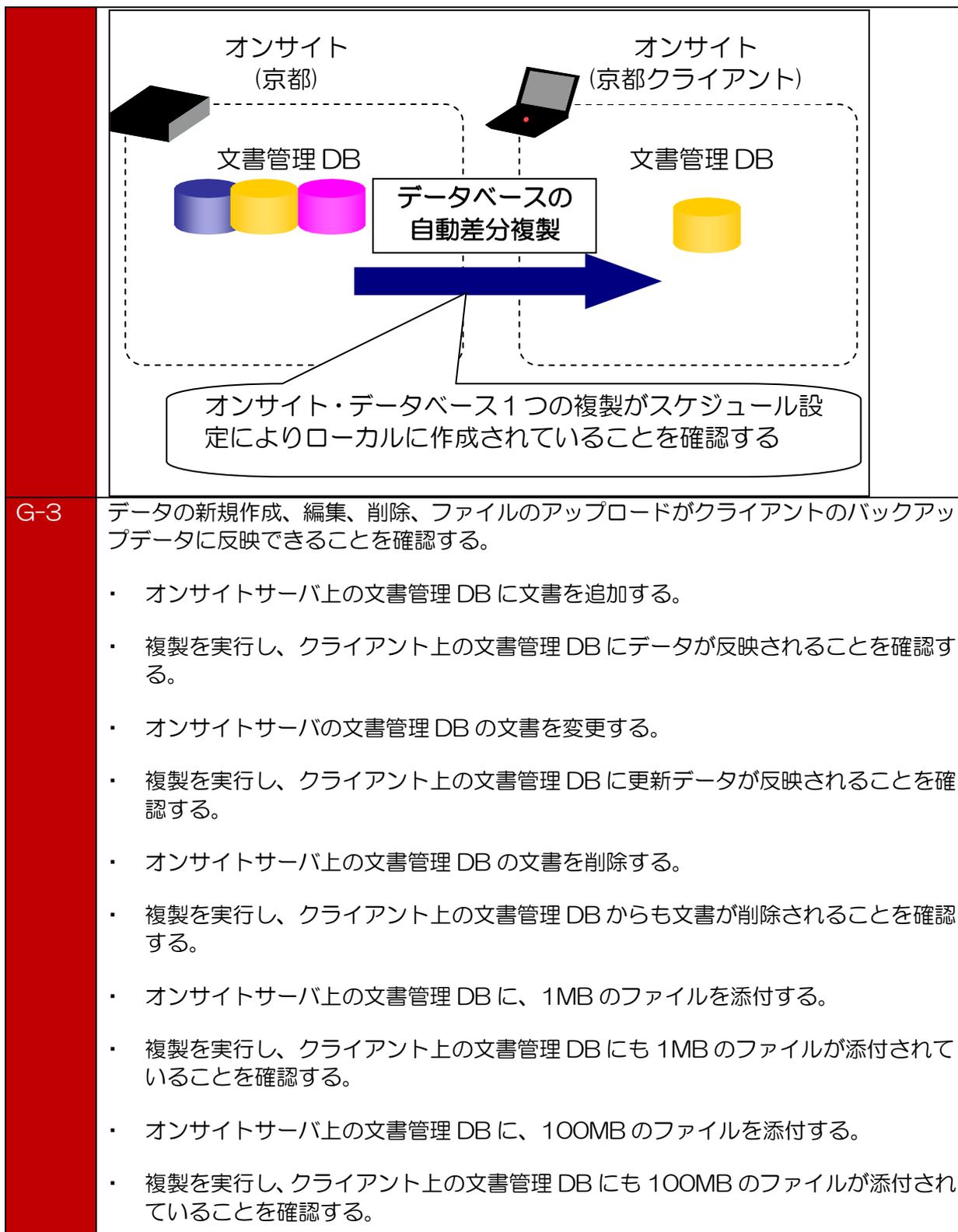
E-3 オンサイト復旧時に、オンサイトでの業務引き戻しが可能なことを確認する

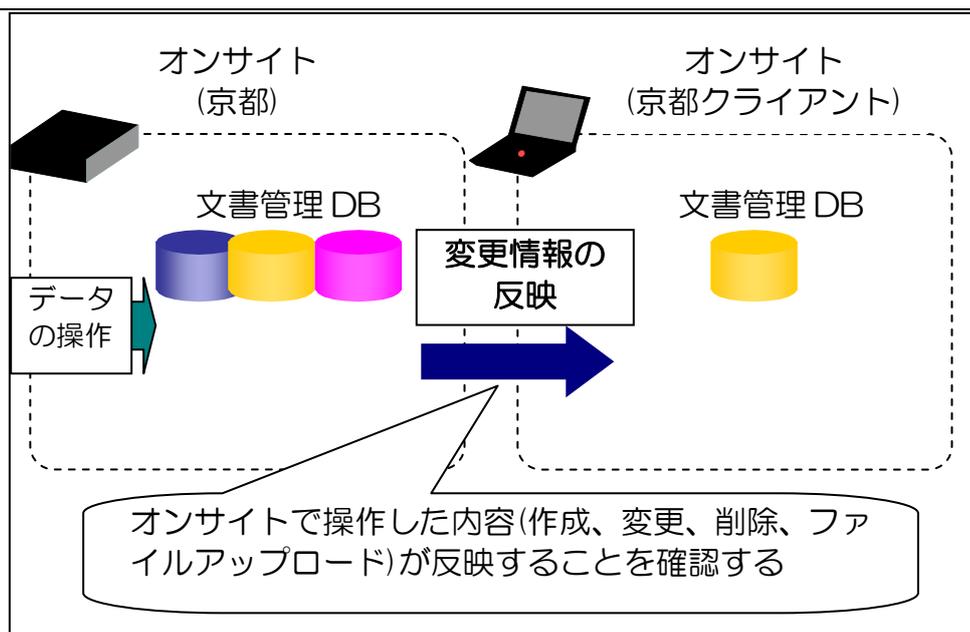
- ・ オンサイトの文書管理 DB、基幹系 DB を削除
- ・ オンサイトに保管しておいた、バックアップデータより文書管理 DB、基幹系 DB を復元
- ・ レプリカ機能および HADR サービスにより、差分データをオフサイトから復旧。
- ・ オンサイトの復旧状況を確認する。



F	メンテナンス時の業務継続性
F-1	<p>セキュリティ対応などの必要な製品修正（Fix Pack）の適用時に、業務を停止せずメンテナンスを実施できることを確認する。</p> <ul style="list-style-type: none"> <li>・ オフサイトのサーバを停止する。</li> <li>・ オンサイトでデータを追加する。</li> <li>・ オフサイトのサーバに修正モジュールを適用する。</li> <li>・ オフサイトのサーバを起動し、修正モジュールのバージョンが異なっても、オンサイトで作業していたデータが同期されることを確認する。</li> </ul> <div data-bbox="344 734 1433 1361" style="border: 1px solid black; padding: 10px;"> <p style="text-align: center;">オフサイトが製品修正を適用した状態でも、データの同期が取られることを確認する。</p> </div>
G	クライアントマシンを利用したバックアップ・リストア
G-1	<p>データを個別にクライアントに、バックアップできることを以下手順にて確認する。</p> <ul style="list-style-type: none"> <li>・ オンサイトの文書管理 DB を選択し、手動で複製設定を行う。</li> <li>・ クライアントに複製する。</li> <li>・ クライアントに複製した文書管理 DB にアクセスし、オンサイトの文書管理 DB と同様の内容であることを確認する。</li> </ul>







また、オンサイトサーバでの削除情報が、クライアントの文書管理 DB に伝播しない設定を以下手順にて確認した。

- ・ オンサイトサーバ上の文書管理 DB で、削除情報を反映しない設定を行う。
- ・ オンサイトサーバ上で文書を作成する。
- ・ 複製を実行し、クライアント上の文書管理 DB に反映されていることを確認する。
- ・ オンサイトサーバ上で文書を削除する。
- ・ 複製を実行し、クライアント上の文書管理 DB に反映されているか確認する。

G-4

データのアクセス権の設定がクライアントの文書管理 DB に反映できることを以下手順にて確認する。

■文書管理 DB の暗号化による制御

- ・ クライアント上での暗号化設定を行い、他ユーザでアクセスする。
- ・ クライアント上の文書管理 DB は、暗号化により開けないことを確認する。
- ・ オンサイトサーバ上の文書管理 DB のアクセス権を剥奪する。
- ・ 複製を実行し、クライアント上の文書管理 DB にアクセス権を剥奪されたユーザがアクセスできないことを確認する。

■文書管理 DB へのアクセス権

- ・ オンサイトサーバ上の文書管理 DB に読者権限のみを付与する。
- ・ 複製を実行し、クライアントの文書管理 DB に接続し、文書管理 DB は表示されるが、新規文書が作成できないことを確認する。

■文書管理 DB の文書へのアクセス権

- ・ オンサイトサーバ上の文書管理 DB の文書に読者権限を付与する。
- ・ 複製を実行し、クライアントの文書管理 DB に接続し、読者権限を付与したユーザ

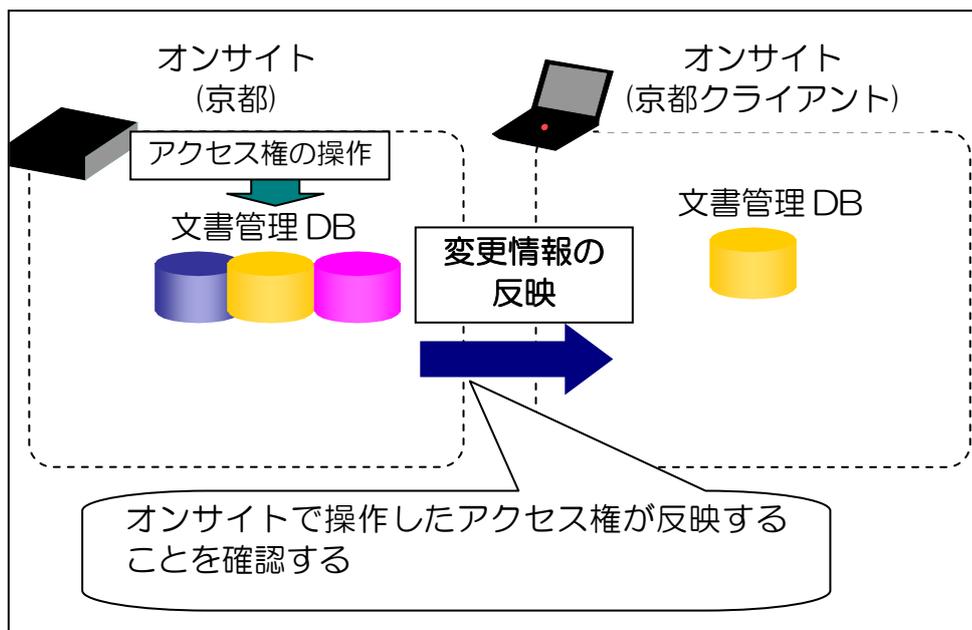
で文書が表示できることを確認する。また、読者権限を付与されていないユーザは、文書自体が表示されないことを確認する。

■文書管理 DB のビューへのアクセス権

- ・ オンサイトサーバの文書管理 DB のビューに読者権限を付与する。
- ・ 複製を実行し、クライアントの文書管理 DB に接続し、読者権限を付与したユーザでビューが表示できることを確認する。また、読者権限を付与されていないユーザは、ビュー自体が表示されないことを確認する。

■文書管理 DB のフィールドへのアクセス権

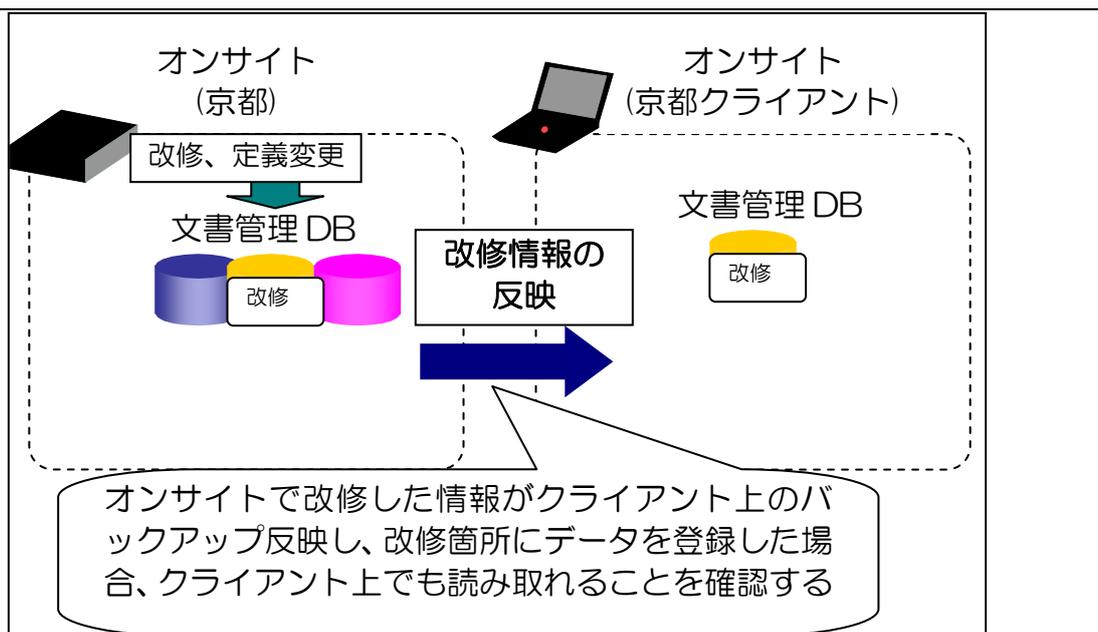
- ・ オンサイトサーバ上の文書管理 DB のフィールドに編集権限を付与する。
- ・ 複製を実行し、クライアントの文書管理 DB に接続し、編集権限を付与したユーザでフィールドが編集できることを確認する。また、編集権限を付与されていないユーザは、閲覧のみとなることを確認する。



G-5

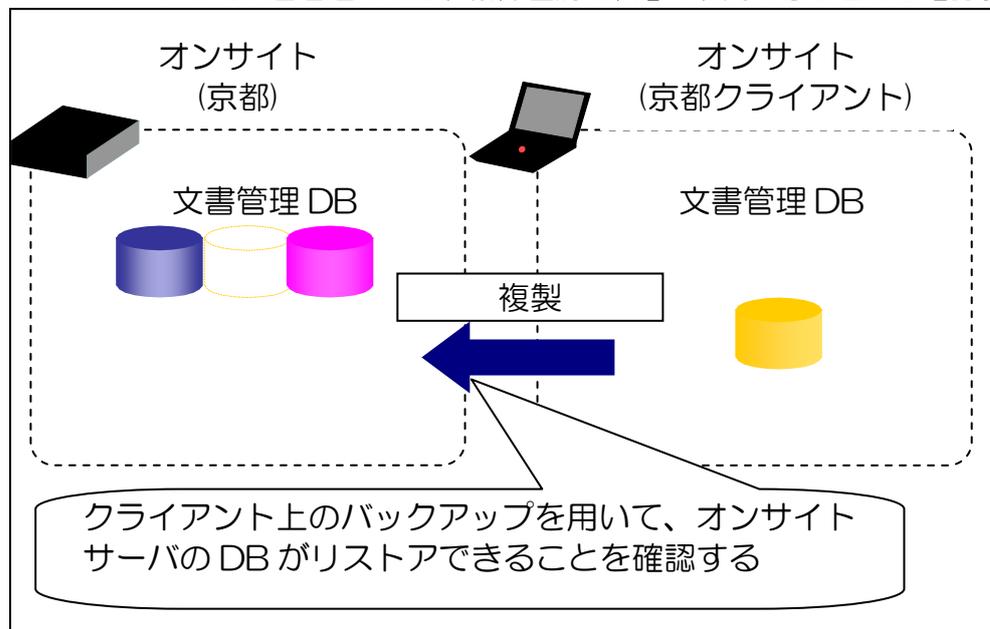
データベースの設計変更がクライアント側へ反映されることを以下手順にて確認する。

- ・ オンサイトサーバ上で文書管理 DB の改修(フィールドの追加)を実施する。
- ・ 改修したフィールドを利用し、文書を作成する。
- ・ クライアント上の文書管理 DB でも同じ改修が反映されていることを確認する。
- ・ 改修したフィールドを利用した文書もクライアント上に反映されていることを確認する。



G-6 クライアントにバックアップしたデータを用いて、リストアできることを確認する。

- ・ オンサイトサーバ上の文書管理 DB を削除する。
- ・ クライアント上の文書管理 DB から、オンサイトサーバ上へ複製の設定を実施する。
- ・ 複製を実行する。
- ・ オンサイトサーバ上の文書管理 DB が、削除直前の状態で利用できることを確認する。



る。

ウ)実証結果

バックアップ・リストアに関するその他有用な機能の実証結果について以下に示す。

E		災害時オフサイト側システムでの業務継続性																																																		
E-1 結果	<p>オフサイトのデータにアクセスした時点から、災害直前までのデータを利用し、オンサイトと同様、文書作成などの業務が継続できることを確認した。文書管理 DB においては、負荷分散装置を配置し、自動切換えを行うことが一般的であり、切り替え時間は一般的には 1 秒程度であるが、今回は負荷分散装置を用意していないため、検証外とする。基幹系 DB においては、プライマリの切り替えに 2 秒を要した。</p>																																																			
E-2 結果	<p>オフサイト・バックアップを業務アプリケーションとして利用した場合のレスポンスについて、オンサイト利用時と比較検証をした。比較検証には、異なるファイルサイズのセキュリティ・パッケージ(10MB、50MB、100MB)を使用し、このセキュリティ・パッケージには、複数の Office 文書が保管されているものとする。また、参考データとして ZIP 圧縮されたデータを利用し、展開の時間を比較測定した。レスポンス測定は以下結果となった。</p> <p>【操作について】</p> <p>サーバ上の文書管理 DB に保管されているセキュリティ・パッケージ化された添付ファイルをダブルクリックすると、自動的にローカルの一時フォルダに保管され、クライアント上で展開用ランチャが起動する。このランチャが起動するまでの時間を”ランチャが開くまでの時間”とする。</p> <p>また、ランチャが起動し、その後「展開」を選択すると、パッケージが展開され、保管されていた複数ファイルが利用できるようになる。このパッケージを展開する時間を”保護フォルダが開くまでの時間”とする。</p> <table border="1"> <thead> <tr> <th>サイズ</th> <th>操作</th> <th>オンサイト</th> <th>オフサイト</th> </tr> </thead> <tbody> <tr> <td rowspan="2">10MB</td> <td>ランチャが開くまでの時間</td> <td>2.07</td> <td>36.03</td> </tr> <tr> <td>保護フォルダが開くまでの時間</td> <td>4.17</td> <td>4.46</td> </tr> <tr> <td rowspan="2">50MB</td> <td>ランチャが開くまでの時間</td> <td>6.04</td> <td>175.32</td> </tr> <tr> <td>保護フォルダが開くまでの時間</td> <td>6.54</td> <td>6.88</td> </tr> <tr> <td rowspan="2">100MB</td> <td>ランチャが開くまでの時間</td> <td>10.70</td> <td>341.86</td> </tr> <tr> <td>保護フォルダが開くまでの時間</td> <td>8.42</td> <td>9.07</td> </tr> </tbody> </table> <p>(表内の数字の単位は秒)</p> <p>比較として、ZIP 圧縮されたファイルの展開は以下となった。</p> <p>【操作について】</p> <p>サーバ上の文書管理 DB に保管された ZIP 圧縮ファイルも、ダブルクリックすると、自動的にローカルの一時フォルダに保管され、クライアント上で展開用ランチャが起動する。このランチャが起動するまでの時間を”ZIP ランチャが開くまでの時間”とする。また、ZIP ランチャが起動し、その後「展開」を選択すると、ZIP ファイルが展開され、保管されていた複数ファイルが利用できるようになる。この ZIP ファイルが展開されるまでの時間を”ZIP ファイルが開くまでの時間”とする。</p> <table border="1"> <thead> <tr> <th>サイズ</th> <th>操作</th> <th>オンサイト</th> <th>オフサイト</th> </tr> </thead> <tbody> <tr> <td rowspan="2">10MB</td> <td>ZIP ランチャが開くまでの時間</td> <td>1.39</td> <td>22.49</td> </tr> <tr> <td>ファイルが開くまでの時間</td> <td>1.89</td> <td>2.12</td> </tr> <tr> <td rowspan="2">50MB</td> <td>ZIP ランチャが開くまでの時間</td> <td>3.89</td> <td>109.42</td> </tr> <tr> <td>ファイルが開くまでの時間</td> <td>11.23</td> <td>11.83</td> </tr> <tr> <td>100MB</td> <td>ZIP ランチャが開くまでの時間</td> <td>9.06</td> <td>291.39</td> </tr> </tbody> </table>					サイズ	操作	オンサイト	オフサイト	10MB	ランチャが開くまでの時間	2.07	36.03	保護フォルダが開くまでの時間	4.17	4.46	50MB	ランチャが開くまでの時間	6.04	175.32	保護フォルダが開くまでの時間	6.54	6.88	100MB	ランチャが開くまでの時間	10.70	341.86	保護フォルダが開くまでの時間	8.42	9.07	サイズ	操作	オンサイト	オフサイト	10MB	ZIP ランチャが開くまでの時間	1.39	22.49	ファイルが開くまでの時間	1.89	2.12	50MB	ZIP ランチャが開くまでの時間	3.89	109.42	ファイルが開くまでの時間	11.23	11.83	100MB	ZIP ランチャが開くまでの時間	9.06	291.39
サイズ	操作	オンサイト	オフサイト																																																	
10MB	ランチャが開くまでの時間	2.07	36.03																																																	
	保護フォルダが開くまでの時間	4.17	4.46																																																	
50MB	ランチャが開くまでの時間	6.04	175.32																																																	
	保護フォルダが開くまでの時間	6.54	6.88																																																	
100MB	ランチャが開くまでの時間	10.70	341.86																																																	
	保護フォルダが開くまでの時間	8.42	9.07																																																	
サイズ	操作	オンサイト	オフサイト																																																	
10MB	ZIP ランチャが開くまでの時間	1.39	22.49																																																	
	ファイルが開くまでの時間	1.89	2.12																																																	
50MB	ZIP ランチャが開くまでの時間	3.89	109.42																																																	
	ファイルが開くまでの時間	11.23	11.83																																																	
100MB	ZIP ランチャが開くまでの時間	9.06	291.39																																																	

	ファイルが開くまでの時間	10.49	11.58								
	(表内の数字の単位は秒)										
E-3 結果	オンサイト復旧後に、再度オンサイトにアクセスし、災害中オフサイトで追加したデータが反映された状態で、オンサイトにて業務ができることを確認した。また引き戻し操作には、基幹系 DB におけるプライマリの変更に 2 秒、その後のデータ同期は、データ更新量を 1KB 以下としたため、1 秒程度で引き戻しが完了した。										
F	<b>メンテナンス時の業務継続性</b>										
F-1 結果	<p>オフサイトへのバックアップを一時停止し、オフサイトのサーバに修正モジュールを適用した。</p> <p>その後、オフサイトへのバックアップを有効にすると、オンサイトでの変更データを反映し、修正モジュールバージョンが異なっても、データ自体の同期に問題がないことを確認した。</p> <p>また、以下手順を踏むことで、業務を継続したまま段階的にオフサイト、オンサイトともに修正モジュールが適用できることを確認した。</p> <ul style="list-style-type: none"> <li>・ オフサイトの製品修正モジュールを適用し、同期を取った後、ユーザーアクセスをオフサイトに切り替える。</li> <li>・ オンサイトのサーバを停止する。</li> <li>・ オンサイトに修正モジュールを適用する。</li> <li>・ オンサイトを起動し、オフサイトと同期を取る。</li> <li>・ ユーザーアクセスをオンサイトに切り替える。</li> </ul>										
G	<b>クライアントマシンを利用したバックアップ・リストア</b>										
G-1 結果	4MB の文書管理 DB において、手動で複製を実行することにより、オンサイトのクライアントに 9 秒でバックアップが取得できることを確認した。										
G-2 結果	<p>以下のスケジュール設定による、クライアントへのバックアップ状況を確認した。</p> <table border="1"> <thead> <tr> <th>スケジュール設定</th> <th>バックアップされたタイミング</th> </tr> </thead> <tbody> <tr> <td>3 分毎でのバックアップ</td> <td>3 分ごとにバックアップデータが更新されることを確認</td> </tr> <tr> <td>クライアント起動時にバックアップ</td> <td>クライアント起動時にバックアップが更新されることを確認。以降サーバのデータを更新しても、クライアントのバックアップは更新されないことを確認</td> </tr> <tr> <td>クライアント停止時にバックアップ</td> <td>サーバでデータを更新しても、バックアップに反映されないことを確認。クライアントを停止するタイミングで、バックアップの取得が実行されることを確認。</td> </tr> </tbody> </table> <p>また一度複製したデータに対しては、その後差分のみの複製のため、どのスケジュール設定でも、軽いデータであれば 1 秒以内にクライアントへ反映された。1 秒以内に、反映できないデータについては、G-3 にて検証する。</p>			スケジュール設定	バックアップされたタイミング	3 分毎でのバックアップ	3 分ごとにバックアップデータが更新されることを確認	クライアント起動時にバックアップ	クライアント起動時にバックアップが更新されることを確認。以降サーバのデータを更新しても、クライアントのバックアップは更新されないことを確認	クライアント停止時にバックアップ	サーバでデータを更新しても、バックアップに反映されないことを確認。クライアントを停止するタイミングで、バックアップの取得が実行されることを確認。
スケジュール設定	バックアップされたタイミング										
3 分毎でのバックアップ	3 分ごとにバックアップデータが更新されることを確認										
クライアント起動時にバックアップ	クライアント起動時にバックアップが更新されることを確認。以降サーバのデータを更新しても、クライアントのバックアップは更新されないことを確認										
クライアント停止時にバックアップ	サーバでデータを更新しても、バックアップに反映されないことを確認。クライアントを停止するタイミングで、バックアップの取得が実行されることを確認。										

G-3  
結果

オンサイトサーバー上の文書管理 DB への、以下の操作がローカルのバックアップに反映されることを確認した。

	操作時間	データの容量	反映時間
文書管理 DB へのデータ作成	3 秒	1KB 未満	1 秒以内
文書管理 DB へのデータ更新	2 秒	1KB 未満	1 秒以内
文書管理 DB へのデータ削除	1 秒	1KB 未満	1 秒以内

削除情報を反映しない設定を行い、クライアント上のバックアップへの反映状況を以下手順にて確認した。設定や文書削除などの処理は手動で行い、2,3 秒で完了する程度の操作である。

- ・ クライアント上の文書管理 DB において、削除情報を反映しない設定を行う。
- ・ サーバ上の文書管理 DB で文書を削除する。
- ・ サーバ上の文書管理 DB で文書を新規作成する。
- ・ クライアント上の文書管理 DB で、新規作成した文書と、オンサイトでは削除した文書の両方が残っていることを確認した。

文書管理 DB へのファイルアップロードについては、ファイルの容量およびネットワーク性能に応じて、クライアント上の文書管理 DB への反映に一定の時間がかかることを確認した。今回の環境においては、以下結果となった。また、オフサイトでの反映時間と比較しておく。

ファイルサイズ	オンサイト・バックアップへの反映時間	オフサイト・バックアップへの反映時間
1 MB	1 秒	5 秒
100MB	10 秒	10 分

G-4  
結果

オンサイトサーバー上の文書管理 DB へのアクセス制御設定変更が、クライアントのバックアップに反映されることを以下により確認した。アクセス制御設定変更操作は 3 秒程度で完了する。

また、操作後は常に複製を実行するものとし、アクセス権の設定変更を実施したユーザを対象ユーザ、その他ユーザを一般ユーザとする。

操作	複製前		複製後	
	対象ユーザ	一般ユーザ	対象ユーザ	一般ユーザ
クライアント上の DB を対象ユーザの ID で暗号化し、対象ユーザ以外のアクセスを制限	アクセス可能	アクセス可能	アクセス可能	アクセス不可
暗号化解除後、オンサイトの文書管理 DB のアクセス権剥奪	アクセス可能	アクセス可能	アクセス不可	アクセス可能
オンサイトの文書管理 DB に読者権限を付与	アクセス不可	編集まで可能	閲覧のみ可能	編集まで可能
オンサイトの文書管理 DB の文書に読者権限にて制限	対象文書が表示	対象文書が表示	対象文書の閲覧が可能	対象文書が非表示
オンサイトの文書管理 DB のビューを読者権限にて制限	対象ビューが表示	対象ビューが表示	対象ビューの閲覧が可能	対象ビューが非表示
オンサイトの文書管理 DB のフィールドを読者権限にて制限	対象フィールドの編集可能	対象フィールドの編集が可能	対象フィールドの閲覧が可能	対象フィールドが非表示

上記より、権限がないユーザは、対象の文書やビューが非表示となり、閲覧ができない結果となった。

	また、クライアント・バックアップへの設定変更反映時間を以下に示す。	
	操作	反映時間
	オンサイトの文書管理 DB のアクセス権を剥奪	1 秒以内
	オンサイトの文書管理 DB に読者権限を付与	1 秒以内
	オンサイトの文書管理 DB の文書に読者権限にて制限	1 秒以内
	オンサイトの文書管理 DB のビューを読者権限にて制限	1 秒以内
G-5 結果	オンサイトの文書管理 DB で、フィールドの追加を実施し、追加したフィールドを利用し、文書を作成すると、クライアント上のバックアップにも、新しいフィールドを利用した文書が登録されていることが確認できた。 また、オンサイトでの文書作成後の複製時点より、1 秒以内に、クライアント上のバックアップに新規文書が登録されていることを確認できた。	
G-6 結果	クライアントからの複製を実行することにより、19MB の文書管理 DB が 9 秒でオンサイトサーバー上にリストアできることを確認した。	

## Ⅰ) 実証の考察

### A. オフサイトを利用した業務の継続性について

E-1 のリアルタイムでデータの同期が取られているため、バックアップデータを利用して、業務が継続できることが確認できた実証結果より、職員にとっては、オフサイトのバックアップを利用して機能面では全く同じアプリケーションを利用する状況になる。ただし、以下結果より、データ容量の大きいファイルを取り扱う場合、アプリケーションには依存せず、バックアップへの反映や、オフサイトからのダウンロードに時間がかかっていることがわかる。よってネットワークやマシンに依存して、オフサイトの業務アプリケーション利用の操作に時間のかかる可能性がある。オフサイトのバックアップデータを一時的に業務で使用するためには、オフサイトシステムにも、相応のマシンリソースやディスク構成、データ領域が必要になる。

表 4-52 検証結果比較

比較項目	所要時間	検証番号
100MB のファイルのオフサイトバックアップへの反映時間	10 分 12 秒	A-3
100MB のファイルのオンサイトバックアップへの反映時間	10 秒	G-3
オフサイトでの 10MB のセキュリティ・パッケージ展開(ランチャ起動まで)	36.03 秒	E-2
オンサイトでの 10MB のセキュリティ・パッケージ展開(ランチャ起動まで)	2.07 秒	E-2
オフサイトでの 10MB の ZIP ファイル展開(ランチャ起動まで)	22.49 秒	E-2
オンサイトでの 10MB の ZIP ファイル展開(ランチャ)	1.39 秒	E-2

起動まで)		
-------	--	--

また、復旧時間については、今回の実証では災害中のオフサイトでのデータ更新を 1KB 程度としたため、プライマリの切り替えに 2 秒、オフサイトのデータとの同期に 1 秒の計 3 秒を復旧に要した。災害中のデータ更新量、災害継続時間によって復旧時間は比例し長くなることが想定される。

#### B. メンテナンス時の業務継続性について

F-1 の製品に修正モジュールを適用する場合において、多少オフサイト、オンサイトで製品修正モジュールのバージョンが異なっても、データの同期には問題がないことが確認できた。この結果より、各自治体が一括で修正モジュールを適用することが難しい場合でも、段階的に修正モジュールを適用することができる。この仕組みは、様々な自治体に参加する自治体クラウドにおいて、有効な機能となる。

#### C. 管理者のクライアントへのバックアップ・リストアについて

G-1、G-2 の管理者のクライアント上にもオフサイトと同様に、バックアップを取得できるという結果より、突発的な作業前にバックアップを取得しておく(例えば文書の一括削除作業時において、誤って削除した文書の復旧用や設計変更を実施したいときの既存環境への影響確認用)など、管理者が任意の時点でのバックアップを手元に保管することができる。自治体クラウド環境でも、要件によって各自治体がバックアップ運用を検討することができ有効となる。また、以下結果より短時間でバックアップ・リストアしたい場合には、オンサイトでのバックアップが有効である。

表 4-53 バックアップ時の所要時間

操作	所要時間	検証番号
オフサイトへの 19MB の文書管理 DB バックアップ	1 分 35 秒	A-2
オンサイトへの 19MB の文書管理 DB バックアップ	9 秒	G-1

表 4-54 リストア時の所要時間

操作	所要時間	検証番号
オフサイトからの 19MB の文書管理 DB リストア	20 秒	C-1
オンサイトからの 19MB の文書管理 DB リストア	9 秒	G-6

#### D. 管理者のクライアント上のバックアップデータの反映について

オフサイトバックアップと同様、以下結果より管理者のクライアント上にあるバックアップデータに各種データを反映させることができることが実証できた。クライアント上のバックアップも、オフサイトのバックアップと同様正確な情報をバックアップできる仕組みとして有効である。

表 4-55 検証結果

操作	サーバとクライアントの内容比較	反映時間	検証番号
文書管理 DB でのデータの新規追加	サーバと同様	1 秒	G-3
文書管理 DB でのデータの変更	サーバと同様	1 秒	G-3
文書管理 DB でのデータの削除	サーバと同様	1 秒	G-3
文書管理 DB でのアクセス権	サーバと同様	1 秒	G-4
文書管理 DB に添付したセキュリティ・データ(1MB)	サーバと同様	1 秒	G-3
文書管理 DB に添付したセキュリティ・データ(100MB)	サーバと同様	10 秒	G-3

## (6) 障害時のバックアップサーバへの切替え

### ア) オフサイトバックアップ実証

#### A. 実証の概要・目的

##### a. 目的

地域災害等の発生時において業務継続性を確保するためには、業務データを定期的に遠隔拠点へバックアップしておくこと（オフサイトバックアップという。）が有効とされている。

本実証実験は、総合行政ネットワーク（L GWAN）上でオフサイトバックアップを実施し、業務を継続するための機能性及び実用性について確認することを目的とした。

##### b. 概要

京都府データセンターと北海道データセンターにそれぞれメールサーバを構築し、拠点間でメールデータのオフサイトバックアップを行う。バックアップは、いわゆる通常のファイルバックアップではなく、差分情報を短間隔で定期的に伝送する方式（以下、「オンライン同期方式」という）で実施することとする。オンライン同期方式は、通常のファイルバックアップによるオフサイトバックアップに比して次のような利点がある。本実証では、このオンライン同期方式の有用性についても併せて確認した。オンライン同期方式は、Exchange Server 2010に搭載されている機能（Database Available Group）を利用する。

##### c. オンライン同期方式の利点

- 一般的なオフサイトバックアップは、半日または日単位でデータを伝送するため、地域災害等によるシステム障害が発生した場合に、データの損失量が多くなる可能性が高い。一方、オンライン同期方式では、差分情報を短間隔で伝送するためデータの損失量は少なくなる。
- 片方の拠点のメールサーバに障害が発生した後に、簡易な操作で正常に稼働しているメールサーバに処理が切り替わるように構成することが出来る。そのため、復旧時間の短縮が期待され、エンドユーザーに対する影響が少なくなる。

#### B. 実証の内容

##### a. 実施にあたっての条件

- 京都府データセンター及び北海道データセンター間のL GWANにおいてVPN接続が確立していること。
- 京都府データセンター及び北海道データセンターにおいて使用可能な

物理サーバはそれぞれ1台ずつ。

- すべての仮想マシンは事業者の実証環境にて作成済み。また、必要なソフトウェア等のインストールや諸設定についても事前実施済み。
- 実証実験環境のネットワーク構成上の制約により、クライアントPC(仮想マシン)は庁内LANではなく、京都府データセンターの仮想化サーバと同じセグメントに配置。

b. 利用したツール/ソフトウェア等

利用したツール/ソフトウェア等
<ul style="list-style-type: none"><li>• Microsoft Windows Server 2008 R2 Hyper-V (仮想化サーバ)</li><li>• Active Directory Domain Services (認証サーバ)</li><li>• Microsoft Exchange Server 2010 (メッセージング・グループウェアサーバ)</li><li>• Windows 7 Enterprise (クライアント)</li><li>• Microsoft Outlook 2010 (メールクライアント)</li></ul>

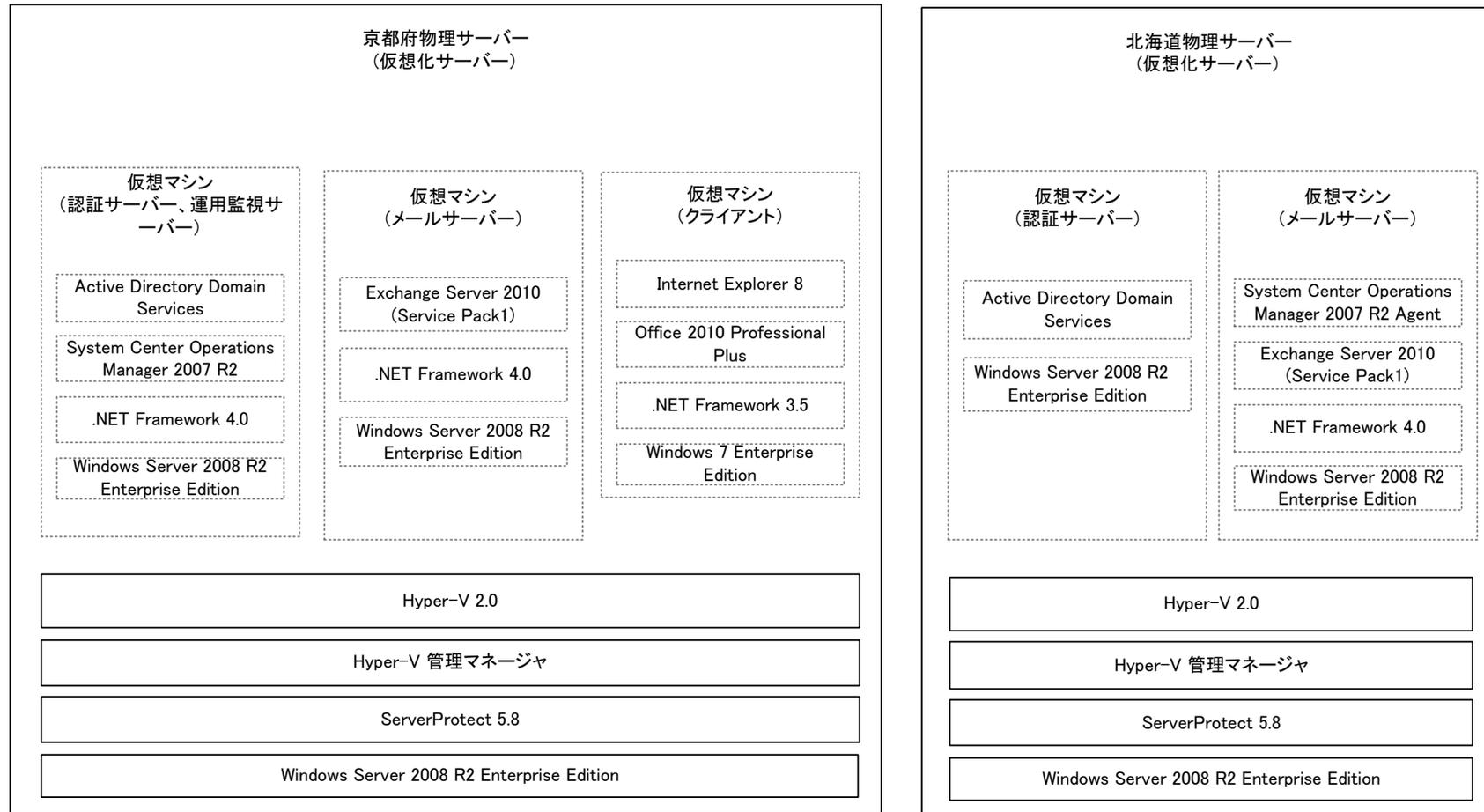


図 4-40 オフサイトバックアップ実証におけるサーバ構成 (京都府データセンター・北海道データセンター)

c. 実証方法

A) オフサイトバックアップ実証

総合行政ネットワーク（L GWAN）上でオフサイトバックアップを実施し、問題なく利用出来ることを確認するため、以下の実証実験を実施した。実証実験の大まかな流れを下表に示す。

表 4-56 オフサイトバックアップ実証内容

No.	実施内容
1	実証実験に使用する仮想環境を構築する。
2	クライアントから京都府メールサーバにアクセス出来ることを確認する。
3	クライアントからメールを送信する。
4	仮想環境のメールサーバで意図的に障害を発生させる。 （仮想マシンを停止する。）
5	京都府メールサーバから北海道メールサーバに処理を切替えるため、コマンドを実行する。
6	メールクライアントを再起動して北海道メールサーバにアクセスする。
7	事前に送信したメールを確認する。
<b>No.3 のメール送信は複数のパターンにて実施する。No.3～7 の手順を繰り返し実施する。</b>	

- 運用性を確認するための実験環境として、京都府データセンターの仮想化サーバ上に仮想環境の京都府メールサーバを構築した。同じく、北海道データセンターの仮想化サーバ上に仮想環境の北海道メールサーバを構築した。

表 4-57 実証の位置付け

仮想マシン	実証実験における位置付け
京都府メールサーバ	京都府データセンターの仮想化サーバ上で稼働するメールサーバがインストールされた仮想マシン。本実証実験シナリオでは、京都府メールサーバに障害が発生することを想定して実施する。
京都府クライアント	京都府メールサーバを利用するエンドユーザーの端末。
北海道メールサーバ	北海道データセンターの仮想化サーバ上で稼働するメールサーバがインストールされた仮想マシン。本実証実験シナリオでは、京都府メールサーバの待機系（副系）サーバである。

L GWANにおけるオフサイトバックアップの技術実証及び有用性の確認のため、障害発生以前のメールサーバ及びクライアントの正常性を確認した。京都府メールサーバ及び京都府メールサーバとクライアントとの通信が、正常であることを確認するため、以下について確認した。

表 4-58 確認項目

No.	確認項目
1	実証実験環境に仮想マシンのインポートが完了した時点において、仮想マシンの設定画面に表示されている値が、事業者の実証環境において設定したものと同一値であるか。
2	仮想マシンの起動前後に、仮想化サーバ上のイベントログに仮想マシンに関連する“警告”または“エラー”のログが表示されていないか。
3	仮想マシンの起動後に、仮想マシン上のイベントログに“警告”または“エラー”が表示されていないか。
4	エラーメッセージが表示されずに、メールクライアント <sup>2</sup> が起動するか。
5	メールクライアントの接続状況が、“オンライン”と表示されているか。
6	自分宛（ログインユーザー）にテスト メール <sup>3</sup> を送信出来るか。
7	自分宛（ログインユーザー）のテスト メールを受信出来るか。

オンライン同期方式による差分情報の伝送が正常に動作することを確認するため、障害発生前にメールを送信した。本実証実験で使用する Exchange Server 2010 のオンライン同期は、1 MB 単位で差分ログデータをコピーする仕様であることから、機能実証に必要なデータサイズを 1 MB 及び 5 MB とした。メール送信パターンは下表のとおりである。

表 4-59 メールの送信パターン

No.	メール送信パターン	データサイズ
1	テストユーザー1 からテストユーザー2 に対して、1 MB のファイルを添付したメールを 1 通送信する。	1 MB
2	テストユーザー1 からテストユーザー2 に対して、1 MB のファイルを添付したメールを 5 通送信する。	5 MB

メール送信後、LGWAN に接続された京都府メールサーバと北海道メールサーバにおいて、オンライン同期方式が正常に稼働していることを確認するため、以下について確認した。下表の確認項目は、京都府メールサーバ障害発生前に実施した。

<sup>2</sup> Microsoft Outlook 2010

<sup>3</sup> テストメールの内容は次のとおり。（メール件名：テスト メール本文：テスト 添付ファイル：なし テキスト形式）

表 4-60 メール送信パターンNo. 1の確認項目

No.	確認項目
1	メール送信後に京都府メールサーバのデータベースログが作成されるフォルダ <sup>4</sup> に同期用のファイルが作成されるか。
2	京都府メールサーバの同期用のファイルと同数のファイルが、北海道メールサーバのデータベースログが格納される既定のフォルダにコピーされるか。
3	京都府メールサーバ及び北海道メールサーバのイベントログにオンライン同期に関連する“警告”または“エラー”のログが表示されていないか。

表 4-61 メール送信パターンNo. 2の確認項目

No.	確認項目
1	メール送信後に京都府メールサーバのデータベースログが作成されるフォルダに同期用のファイルが作成されるか。
2	京都府メールサーバの同期用のファイルと同数のファイルが、北海道メールサーバのデータベースログが格納される既定のフォルダにコピーされるか。
3	京都府メールサーバ及び北海道メールサーバのイベントログにオンライン同期に関連する“警告”または“エラー”のログが表示されていないか。

京都府メールサーバに障害が発生している状況を用意するため、仮想化ソフトウェアの管理ツールを用いて、京都府メールサーバの仮想マシンを停止（シャットダウン）した。京都府メールサーバの仮想マシンを停止したことで、京都府メールサーバとクライアントとの通信が途絶し、メールアイテムにアクセス出来ないことを確認するため、以下について確認した。

表 4-62 メール送信パターンNo. 1の確認項目

No.	確認項目
1	メールクライアントの起動時に、「サーバは利用出来ません」とメッセージダイアログが表示され、起動に失敗するか。
2	北海道メールサーバの管理コンソールで、京都府メールサーバのメールデータベースの接続情報を参照した際に、「インフォメーションストアサービスに接続出来ません」と警告メッセージが表示されるか。
3	北海道メールサーバの管理コンソールで、京都府メールサーバのデータベースコピーのステータスを参照した際に、“サービス停止”と表示されるか。

表 4-63 メール送信パターンNo. 2の確認項目

No.	確認項目
1	メールクライアントの起動時に、「サーバは利用出来ません」とメッセージダイアログが表示され、起動に失敗するか。
2	北海道メールサーバの管理コンソールで、京都府メールサーバのメールデータベースの接続情報を参照した際に、「インフォメーションストアサービスに接続出

<sup>4</sup> フォルダのパスは、次のとおり。 C:\Program Files\Microsoft\Exchange Server\V14\Mailbox\MDB01

	来ません」と警告メッセージが表示されるか。
3	北海道メールサーバの管理コンソールで、京都府メールサーバのデータベースコピーのステータスを参照した際に、“サービス停止”と表示されるか。

京都府メールサーバの仮想マシンを停止後、北海道メールサーバにフェールオーバークラスターサービスを強制的に認識させるため、北海道メールサーバでプロンプトを起動し、フェールオーバークラスターサービスを再起動するコマンドを実行した。実行したコマンドについて下表に示す。

表 4-64 実行コマンド

実行したコマンド	コマンド処理内容
NET STOP CLUSSVC	(オンライン同期方式に必要な)フェールオーバークラスターサービスを停止する。
NET START CLUSSVC /fq	(オンライン同期方式に必要な)フェールオーバークラスターサービスを強制的に開始する。

コマンドが正常に実行されたことを実証するため、以下について確認した。

表 4-65 確認項目

No.	確認項目
1	コマンド実行後、コマンドプロンプトにそれぞれ次のメッセージが表示されるか。  サービス停止時：「Cluster Service サービスは正常に停止されました」 サービス開始時：「Cluster Service サービスは正常に開始されました」
2	北海道メールサーバの管理コンソールで、北海道メールサーバのデータベースコピーのステータスを参照した際に、“マウント済み <sup>5)</sup> ”と表示されるか。

クライアントの接続先を、停止した京都府メールサーバから、北海道メールサーバに切り替えるため、北海道メールサーバでメールサーバの管理用コマンドシェル<sup>6)</sup>を起動し、コマンドを実行した。実行したコマンドについて下表に示す。

表 4-66 実行コマンド

実行したコマンド	コマンド処理内容
Set-MailboxDatabase MDB01 -RpcClientAccessServer EX-HOK-V.cloud-test.local	クライアントの接続先サーバを “EX-HOK-V” (北海道サーバのホスト名) に切替える。

<sup>5)</sup> メールサーバーにメールボックス データベースを認識させ、操作可能にすることを「マウントする」という。ここでは、京都府メールサーバーにマウントされていたメールボックスデータベースコピーを、北海道メールサーバーで正常に認識した状態を指す。

<sup>6)</sup> Exchange Management Shell

コマンドが正常に実行されたことを実証するため、以下について確認した。

表 4-67 確認項目

No.	確認項目
1	コマンドの実行後、コマンドシェルにエラーメッセージが表示されないか。
2	クライアントの接続先サーバの詳細ステータスを取得する次のコマンド <pre>Get-MailboxDatabase MDB01   Format-List</pre> を実行し、RpcClientAccessServer の値が、"EX-HOK-V"（北海道サーバのホスト名）であるか。

#### B) 障害発生時におけるクライアント操作実証

メールクライアントには接続先のメールサーバの情報が格納されるため、メールクライアントのアカウント設定で接続先サーバを変更する必要がある。

障害が発生した京都府メールサーバから北海道メールサーバに接続先を切替えた後、クライアント側で必要な操作について実証するため、テストユーザー2でログインしたクライアントで、メールクライアントを起動し、自動アカウント設定ウィザードを実行することで北海道メールサーバに再接続が可能であるか下表の確認項目について確認した。

表 4-68 確認項目

No.	確認項目
1	メールクライアント起動後に「サーバは利用出来ません」のメッセージダイアログが表示され、「再試行」を選択した際に、オプション画面に遷移するか。
2	オプション画面にて「アカウント設定」を選択し、続けて「修復」を選択した際に、アカウントの修復ウィザードが起動するか。
3	アカウントの修復ウィザードの電子メールアカウント情報を入力するテキストボックスにログインユーザーであるテストユーザー2の名前とメールアドレスがそれぞれ自動設定されるか。
4	アカウントの修復ウィザードで「サーバ設定のオンライン検索処理」の以下の処理が正常に完了したか。 ネットワーク接続の確立 サーバ設定の検索 サーバへのログオン
5	続けて、「変更を有効にするには、Outlook を再起動する必要があります。」のメッセージダイアログが表示されるか。
6	アカウントの修復ウィザードが終了し、「電子メールアカウントの設置が完了しました。」とメッセージが表示されるか。

No.	確認項目
7	メールクライアント を再起動後、ステータスバーの“Microsoft Exchange の接続状態”を参照し、動作状況タブに表示される接続先のサーバが、北海道メールサーバ（EX-HOK-V.cloud-test.local）となっているか。

再接続に成功した後、京都府メールサーバの障害発生前に送信したメールアイテムを北海道メールサーバに接続して閲覧出来るか確認した。

表 4-69 メール送信パターンNo. 1の確認項目

No.	確認項目
1	京都府メールサーバの障害発生以前に送信した 1MB のファイルを添付した 1 通のメールを北海道メールサーバに接続した際に閲覧出来るか。

表 4-70 メール送信パターンNo. 2の確認項目

No.	確認項目
1	京都府メールサーバの障害発生以前に送信した 1MB のファイルを添付した 5 通のメールを北海道メールサーバに接続した際に閲覧出来るか。

### C. 実証の結果

総合行政ネットワーク（LGWAN）上でオフサイトバックアップを実施し、問題なく利用出来ることを実証するため、以下の実証実験を行い、次の結果が得られた。

#### a. オフサイトバックアップの実証結果

環境は事業者の検証環境で作成した仮想マシンを使用することで本実証実験を実施した。

LGWANにおけるオフサイトバックアップの技術実証及び有用性の確認のため、障害発生以前のメールサーバ及びクライアントの正常性について確認した。

実証の結果については下表のとおりである。

表 4-71 実証の結果

No.	確認項目	実証の結果
1	実証実験環境に仮想マシンのインポートが完了した時点において、仮想マシンの設定画面に表示されている値が、事業者の実証環境において設定したものと同一値であるか。	京都府メールサーバ、北海道メールサーバ、認証サーバ及びクライアントの、すべての仮想マシンの設定値において事業者の実証環境にて設定した値と同一であった。
2	仮想マシンの起動前後に、仮想化サーバ上のイベントログに仮想マシ	京都府仮想化サーバ及び北海道仮想化サーバのイベントログにおいて、仮想マシンに関する

	ンに関連する“警告”または“エラー”のログが表示されていないか。	“警告”、または“エラー”のログが表示されることなく起動・動作した。
3	仮想マシンの起動後に、仮想マシン上のイベントログに“警告”または“エラー”が表示されていないか。	京都府メールサーバ、北海道メールサーバ、認証サーバ及びクライアントの仮想マシンのイベントログにおいて、仮想マシンに関する“警告”、または“エラー”のログが表示されることなく起動・動作した。
4	エラーメッセージが表示されずに、メールクライアントが起動するか。	エラーメッセージが表示されることなく、メールクライアントが起動した。
5	メールクライアントの接続状況が、“オンライン”と表示されているか。	メールクライアントの接続状況を示すステータスバーの表示内容が、起動直後の“接続中”から10秒で“オンライン”に変更された。
6	自分宛（ログインユーザー）にテストメールを送信出来るか。	自分宛にテストメールを送信し、送信済みアイテムフォルダにアイテムが移動した。
7	自分宛（ログインユーザー）のテストメールを受信出来るか。	自分宛に送信したテストメールを受信し、メッセージを閲覧出来た。

京都府メールサーバから北海道メールサーバに短時間で差分情報を定期的に伝送する方式が正常に動作することを実証するため障害発生前にメールを送信した。

メールを送信した結果については下表のとおりである。

表 4-72 実証の結果

No.	内容	データサイズ	実証の結果	送信に要した時間 <sup>7</sup>
1	テストユーザー1からテストユーザー2に対して、1MBのファイルを添付したメールを1通送信する。	1MB	送信処理を実行後、送信済みフォルダに格納されたことで正常にメールが送信されたことを確認した。	5秒
2	テストユーザー1からテストユーザー2に対して、1MBのファイルを添付したメールを5通送信する。	5MB	送信処理を実行後、送信済みフォルダに格納されたことで正常にメールが送信されたことを確認した。	20秒

メール送信後、L GWANに接続された京都府メールサーバと北海道メールサーバにおいて、オンライン同期方式が正常に稼働していることを確認した結果は下表のとおりである。下表の確認項目は、京都府メールサーバ障害発生前に実施した。

<sup>7</sup> 送信ボタン押下後、送受信が完了し送信済みフォルダにアイテムが移動するまでに要した時間。

表 4-73 メール送信パターンNo. 1の確認項目

No.	確認項目	実証の結果	要した時間
1	メール送信後に京都府メールサーバのデータベースログが作成されるフォルダ <sup>8</sup> に同期用のファイルが作成されるか。	メール送信後に京都府メールサーバのデータベースログファイルが格納されるフォルダに同期用のファイルが作成されたことを確認した。	メール送信完了後から、京都府メールサーバにデータベースログファイルが作成されるまでに要した時間は <u>10 秒</u> であった。
2	京都府メールサーバの同期用のファイルと同数のファイルが、北海道メールサーバのデータベースログが格納される既定のフォルダにコピーされるか。	京都府メールサーバの同期用のファイルと同数のファイルが、北海道メールサーバのデータベースログが格納される既定のフォルダにコピーされたことを確認した。	メール送信完了後から、北海道メールサーバにデータベースログファイルがコピーされるまでに要した時間は <u>30 秒</u> であった。
3	京都府メールサーバ及び北海道メールサーバのイベントログにオンライン同期に関連する“警告”または“エラー”のログが表示されていないか。	京都府メールサーバ及び北海道メールサーバのイベントログにオンライン同期に関連する“警告”または“エラー”のログが表示されていないことを確認した。	-

メール送信パターンNo. 1において、メール送信後から北海道メールサーバにデータベースログファイルがコピーされ、オンライン同期が完了するまでの30秒間で使用した、北海道メールサーバのネットワーク帯域は、平均値：54.5KB/sec、最大値：156.9KB/secであった。

表 4-74 メール送信パターンNo.2の確認項目

No.	確認項目	実証の結果	要した時間
1	メール送信後に京都府メールサーバのデータベースログが作成されるフォルダに同期用のファイルが作成されるか。	メール送信後に京都府メールサーバのデータベースログファイルが格納されるフォルダに同期用のファイルが作成されたことを確認した。	メール送信完了後から、京都府メールサーバにデータベースログファイルが作成されるまでに要した時間は <u>40 秒</u> であった。
2	京都府メールサーバの同期用のファイルと同数の	京都府メールサーバの同期用のファイルと同数のファ	メール送信完了後から、北海道メールサーバにデ

<sup>8</sup> フォルダのパスは、次のとおり。 C:\Program Files\Microsoft\Exchange Server\V14\Mailbox\MDB01

	ファイルが、北海道メールサーバのデータベースログが格納される既定のフォルダにコピーされるか。	イルが、北海道メールサーバのデータベースログが格納される既定のフォルダにコピーされたことを確認した。	データベースログファイルがコピーされるまでに要した時間は <u>135</u> 秒であった。
3	京都府メールサーバ及び北海道メールサーバのイベントログにオンライン同期に関連する“警告”または“エラー”のログが表示されていないか。	京都府メールサーバ及び北海道メールサーバのイベントログにオンライン同期に関連する“警告”または“エラー”のログが表示されていないことを確認した。	-

メール送信パターンNo. 2において、メール送信後から北海道メールサーバにデータベースログファイルがコピーされ、オンライン同期が完了するまでの135秒間で使用した、北海道メールサーバのネットワーク帯域は、平均値：40.3KB/sec、最大値：793.6KB/secであった。

京都府メールサーバに障害が発生している状況を用意するため、仮想化ソフトウェアの管理ツールを用いて、京都府メールサーバの仮想マシンを停止させた。確認項目に対する実証の結果については下表のとおりである。

表 4-75 実証の結果

No.	確認項目	実証の結果
1	メールクライアントの起動時に、「サーバは利用出来ません」とメッセージダイアログが表示され、起動に失敗するか。	メールクライアントを起動した際に、メッセージダイアログに「サーバは利用出来ません」と表示された。
2	北海道メールサーバの管理コンソールで、京都府メールサーバのメールデータベースの接続情報を参照した際に、「インフォメーションストアサービスに接続出来ません」と警告メッセージが表示されるか。	北海道メールサーバの管理コンソールで、京都府メールサーバのメールデータベースの接続情報を参照した際に、「インフォメーションストアサービスに接続出来ません」と警告メッセージが表示された。
3	北海道メールサーバの管理コンソールで、京都府メールサーバのデータベースコピーのステータスを参照した際に、“サービス停止”と表示されるか。	北海道メールサーバの管理コンソールで、京都府メールサーバのデータベースコピーのステータスを参照した際に、“サービス停止”と表示された。

京都府メールサーバの仮想マシンを停止後、北海道メールサーバにフェールオーバークラスターサービスを認識させるため、北海道メールサーバでコマンドプロンプトを起動、フェールオーバークラスターサービスを再起動するコマンドを実行した。フェールオーバークラスターサービスを再

起動するコマンドの実行が完了するまでに要した時間は、20秒であり、コマンドの実行結果については下表のとおりである。

表 4-76 実証の結果

No.	確認項目	実証の結果
1	<p>コマンド実行後、コマンドプロンプトにそれぞれ次のメッセージが表示されるか。</p> <p>サービス停止時：「Cluster Service サービスは正常に停止されました」</p> <p>サービス開始時：「Cluster Service サービスは正常に開始されました」</p>	<p>コマンド実行後、コマンドプロンプトにそれぞれ次のメッセージが表示されたことを確認した。</p> <p>サービス停止時：「Cluster Service サービスは正常に停止されました」</p> <p>サービス開始時：「Cluster Service サービスは正常に開始されました」</p>
2	<p>北海道メールサーバの管理コンソールで、北海道メールサーバのデータベースコピーのステータスを参照した際に、“マウント済み<sup>9)</sup>”と表示されるか。</p>	<p>北海道メールサーバの管理コンソールで、北海道メールサーバのデータベースコピーのステータスを参照した際に、“マウント済み”と表示されたことを確認した。</p>

クライアントの接続先を、停止した京都府メールサーバから北海道メールサーバに切替えるため、北海道メールサーバでメールサーバの管理用コマンドシェルを起動、コマンドを実行した。京都府メールサーバから、北海道メールサーバに切替えるコマンドの実行が完了するまでに要した時間は8秒であった。確認結果は下表のとおりである。

表 4-77 確認結果

No.	確認項目	実証の結果
1	<p>コマンドの実行後、コマンドシェルにエラーメッセージが表示されないか。</p>	<p>コマンドの実行後、コマンドシェルにエラーメッセージが表示されていないことを確認した。</p>
2	<p>クライアントの接続先サーバの詳細ステータスを取得する次のコマンド</p> <pre>Get-MailboxDatabase MDB01   For mat-List</pre> <p>を実行し、<b>RpcClientAccessServer</b> の値が、“EX-HOK-V”（北海道サーバのホスト名）であるか。</p>	<p>Get-MailboxDatabase MDB01   Format-List</p> <p>の実行結果で <b>RpcClientAccessServer</b> の値が、“EX-HOK-V”と表示されたことを確認した。</p>

京都府メールサーバから北海道メールサーバに接続先が切り替わったこ

<sup>9)</sup> メールボックスが保管されるメールデータベースに接続した状態になると、“マウント済み”と表示される。

とを確認した上で、メールクライアントを起動し、自動アカウント設定ウィザードを実行することで北海道メールサーバに再接続がされるか確認した。確認結果は下表のとおりである。メールクライアントのアカウント設定を自動構成するウィザードを実行し、アカウント構成処理が完了するまでに要した時間は150秒であった。

表 4-78 実証の結果

No.	確認項目	実証の結果
1	メールクライアント起動後に「サーバは利用出来ません」のメッセージダイアログが表示され、「再試行」を選択した際に、オプション画面に遷移するか。	メールクライアント起動後に「サーバは利用出来ません」のメッセージダイアログが表示され、「再試行」を選択した際に、オプション画面に遷移したことを確認した。
2	オプション画面にて「アカウント設定」を選択し、続けて「修復」を選択した際に、アカウントの修復ウィザードが起動するか。	オプション画面にて「アカウント設定」を選択し、続けて「修復」を選択した際に、アカウントの修復ウィザードが起動したことを確認した。
3	アカウントの修復ウィザードの電子メールアドレスを入力するテキストボックスにログインユーザーであるテストユーザー2の名前とメールアドレスがそれぞれ自動設定されるか。	アカウントの修復ウィザードの電子メールアドレスを入力するテキストボックスにログインユーザーであるテストユーザー2の名前とメールアドレスがそれぞれ自動でセットされたことを確認した。
4	アカウントの修復ウィザードで「サーバ設定のオンライン検索処理」の以下の処理が正常に完了したか。 ネットワーク接続の確立 サーバ設定の検索 サーバへのログオン	アカウントの修復ウィザードで「サーバ設定のオンライン検索処理」の以下の処理について、画面上に正常に完了したことを示す緑色のチェックが付加されたことを確認した。 「サーバ設定のオンライン検索処理」のすべての処理が完了するまで、 <u>80秒</u> 要した。 ネットワーク接続の確立 サーバ設定の検索 サーバへのログオン
5	続けて、「変更を有効にするには、Outlook を再起動する必要があります。」のメッセージダイアログが表示されるか。	「変更を有効にするには、Outlook を再起動する必要があります。」のメッセージダイアログが表示されたことを確認した。
6	アカウントの修復ウィザードが終了し、「電子メールアドレスの設置が完了しました。」とメッセージが表示されるか。	アカウントの修復ウィザードが終了し、「電子メールアドレスの設置が完了しました。」とメッセージが表示されたことを確認した。
7	メールクライアント を再起動後、ステータスバーの「Microsoft Exchange の接続状態」を参照し、動作状況タブに表示される接続先のサーバが、北海道メールサーバ (EX-HOK-V.cloud-test.loc	メールクライアント を再起動後、ステータスバーの「Microsoft Exchange の接続状態」を参照し、動作状況タブに表示される接続先のサーバが、北海道メールサーバ (EX-HOK-V.cloud-test.local) となっ

al) となっているか。	ていることを確認した。
--------------	-------------

京都府メールサーバの障害発生前に送信したメールアイテムを、障害発生後に北海道メールサーバで接続し、閲覧することが出来るか確認した。

下表に示す確認項目における実証の結果から、オンライン同期方式は正常に動作しており、かつ、北海道メールサーバに再接続することで、障害発生以前に送信したメールアイテムを閲覧することが可能であることを確認した。

表 4-79 メール送信パターン No.1 の確認項目

No.	確認項目	実証の結果
1	京都府メールサーバの障害発生以前に送信した 1MB のファイルを添付した 1 通のメールを北海道メールサーバに接続した際に閲覧出来るか。	テストユーザー1 からテストユーザー2宛てに送信された、1MB のファイルが添付された 1 通のメールをすべて閲覧することが出来た。

表 4-80 メール送信パターン No.2 の確認項目

No.	確認項目	実証の結果
1	京都府メールサーバの障害発生以前に送信した 1MB のファイルを添付した 5 通のメールを北海道メールサーバに接続した際に閲覧出来るか。	テストユーザー1 からテストユーザー2宛てに送信された、1MB のファイルが添付された 5 通のメールをすべて閲覧することが出来た。

#### D. 結果の考察

実証の結果から得られた内容を踏まえ、その考察について記述する。

##### a. オンライン同期方式によるオフサイトバックアップの機能性・実用性について

いずれのメール送信パターンにおいて、メール送信後、京都府メールサーバに同期用ログが生成された。

【実証結果：1MB → 10秒、5MB → 40秒】

すべてのメール送信パターンにおいて、メール送信後、京都府メールサーバに生成されたログが、自動で北海道メールサーバにコピーされた。

【実証結果：1MB → 30秒、5MB → 135秒】

京都府メールサーバに障害が発生した後、北海道メールサーバ上でのコマンド実行及びクライアント上での簡易なウィザード操作によって、北海道サーバ側に再接続出来た。

【実証結果：1MB、5MBいずれも コマンド実行 → 8秒 + 1MB、5MB いずれもフェールオーバークラスターサービスの再起動 → 20

秒 + アカウント再構成ウィザード実行 → 150秒)】

いずれのメール送信パターンにおいて、京都府メールサーバから北海道メールサーバに切替えた後、クライアントから事前に送信したメールデータを閲覧することが出来た。

以上の結果から、オンライン同期方式によるオフサイトバックアップは正常に動作したことを確認出来た。実証実験に要した時間の積算値<sup>10</sup>は、データ同期において、1MBで40秒、5MBで175秒、障害復旧（サーバー切替え）においてそれぞれ178秒であり、自治体クラウドのデータセンターを日本各地に分散配置し、オフサイトバックアップを実施することで業務継続性を確保する方法のひとつとして機能することを確認した。送信したメールのデータ容量が大きくなることで同期が完了するまでに要する時間が増加していることから、データセンター間のネットワーク帯域を確保することが重要であり、今後、自治体クラウドを整備・構築するにあたって、オンライン同期方式によるオフサイトバックアップを導入する場合には、職員数、メール送受信量及び平均サイズ等を基に必要な帯域を見積もる必要がある。

## イ) データセンター間接続実証（システム運用監視）

### A. 実証の概要・目的

自治体クラウド基盤を構成する技術要素（物理サーバ、仮想サーバ（仮想マシン）、ネットワーク、ソフトウェア等）は複数事業者のサービスの組み合わせとなることから、障害の切り分けは、従来のシステムよりも複雑化する。

自治体クラウド基盤で稼働する多数のサーバ及び仮想マシンの運用作業を効率化するためには、運用監視ソフトウェアを活用することで、自動的に障害を検知することが重要である。

本実証実験は、自治体クラウド基盤上で稼働する仮想マシンに対し、単一拠点からL GWANを経由した遠隔による運用監視を行い、運用監視に関する技術実証及び課題の抽出等を行うことを目的とする。

### B. 実証の内容

#### a. 実施にあたっての条件

- 京都府データセンター及び北海道データセンターにおいて使用可能な物理サーバはそれぞれ1台ずつ。
- すべての仮想マシンは事業者の実証環境にて作成済み。また、必要なソフトウェア等のインストールや諸設定についても事前実施済み。

<sup>10</sup>実証実験の実施内容を積算した時間であり、実運用において障害発生から復旧までに要する時間は、障害を検知してから対応するまでの時間等により変動しうる。

- 事業者の実証環境にて作成した仮想マシンは、事業者の用意したUSB接続タイプのポータブルハードディスクに格納し運搬した。

表 4-81 利用したツール/ソフトウェア等

利用したツール/ソフトウェア等
•Microsoft Windows Server 2008 R2 Hyper-V (仮想化サーバ)
•Active Directory Domain Services (認証サーバ)
•Microsoft System Center Operating Manager 2007 R2 (運用監視サーバ)
•Microsoft Exchange Server 2010 (メールサーバ)

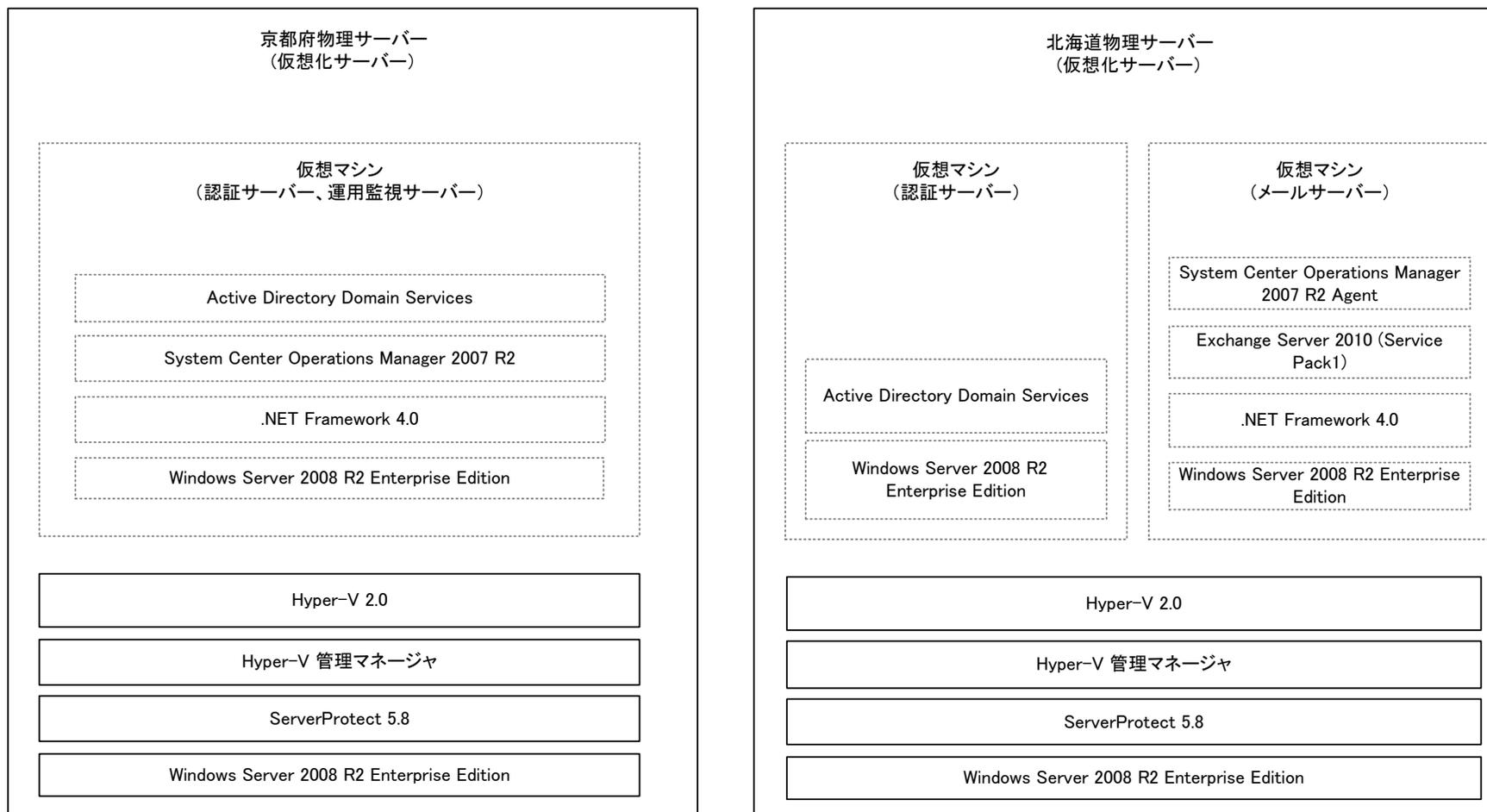


図 4-41 データセンター間接続実証におけるサーバ構成 (京都府データセンター・北海道データセンター)

## b. 実証方法

## A) 運用監視ツールの有効性の実証

複数拠点に分散するデータセンターを一元的に管理し、障害発生時の迅速な障害の切り分けを行うための運用監視ツールの有効性について確認するため、以下の実証実験を実施した。

- 運用性を実証するための実験環境として、京都府データセンターに仮想化サーバを構築し、仮想マシンとして運用監視サーバを構築した。
- 運用監視を行うためには、監視対象のサーバで、自身のコンピュータを監視したり、タスクを実行したりして、その結果を管理サーバに通知するエージェントと呼ばれるプログラムが必要である。京都府データセンターの運用監視サーバから、北海道データセンターのメールサーバに対して、L GWANを経由しリモートでエージェントをインストール出来るか、運用監視サーバの「コンピュータとデバイスの管理ウィザード<sup>11</sup>」を実行した結果について確認した。

表 4-82 確認項目

No.	確認項目
1	「コンピュータとデバイスの管理ウィザード」を実行した結果、“エージェントの管理タスクの状態”に「タスクは正常に完了しました」とメッセージが表示されるか。
2	北海道メールサーバのイベントログにエージェントインストールに“警告”または“エラー”のログが表示されていないか。
3	北海道メールサーバの「インストール済みプログラム一覧」に、エージェントプログラムが表示されているか。
4	運用監視サーバの運用監視コンソールで北海道メールサーバの稼働状態を表示出来るか。

京都府データセンターの運用監視サーバから、L GWANを経由して北海道データセンターのメールサーバを運用監視出来ることを実証するため、北海道メールサーバに接続されているNICを無効化することで、意図的なネットワーク障害に発生させた。

ネットワーク障害を運用監視サーバの運用監視用コンソールで検知出来るか以下の項目について確認した。

表 4-83 確認項目

No.	確認項目
1	北海道メールサーバのNICを無効化した後、運用監視サーバの運用監視用コンソールのアイコン表示色がグリーンからグレーに変更されたか。

<sup>11</sup> ネットワーク上のコンピュータまたはデバイスを検出し、エージェントを配布・インストールするためのウィザード形式のプログラム。

## C. 実証の結果

京都府データセンターの運用監視サーバから、L GWANを経由して北海道データセンターのメールサーバを運用監視出来ることを実証するため、以下の実証実験を実施し、次の結果が得られた。

## a. 統合運用監視ツールの有効性の実証

実証実験に使用する環境は、事業者の検証環境で作成した仮想マシンを使用することで本実証実験を実施した。

運用監視を行うために必要な監視用エージェントを北海道メールサーバにリモートでインストールするため、運用監視サーバの「コンピュータとデバイスの管理ウィザード」を実行した。確認の結果は下表のとおりである。

表 4-84 実証の結果

No.	確認項目	実証の結果
1	「コンピュータとデバイスの管理ウィザード」を実行した結果、“エージェントの管理タスクの状態“に「タスクは正常に完了しました」とメッセージが表示されるか。	以下のエラーメッセージが表示され、インストールが失敗した。  リモート コンピュータ <EX-HOK-V.cloud-test.local> のエージェント管理オペレーションはエージェントのインストールは失敗しました。

ネットワークを経由したエージェントのリモートインストールについて、計3回実施したが、いずれも同様のエラーによって失敗した。運用監視サーバ及び北海道メールサーバの再起動及びそれぞれの仮想マシンのWindowsファイアウォールの無効化を行ったが、エラーを回避出来ず原因を特定出来なかった。

運用監視の実証内容を実施するため、エージェントのリモートインストールは断念し、運用監視サーバのインストールメディアに含まれるローカルインストール用のセットアッププログラムを使用し、ローカルでインストールするよう手順を変更した。確認の結果は下表のとおりである。

表 4-85 実証の結果

No.	確認項目	実証の結果
1	[System Center Operations Manager R2 エージェント セットアップ] ウィザードがエラーが表示されることなく、完了したか。	セットアップウィザードは、エラーが発生することなく完了した。
2	北海道メールサーバのイベントログにエージェントインストールに関連する“警告”または“エラー”のログが表示されていないか。	北海道メールサーバのイベントログにエージェントインストールに“警告”または“エラー”のログが表示されていないかった。
3	北海道メールサーバの「インストール	北海道メールサーバの「インストール

No.	確認項目	実証の結果
	済みプログラム一覧」に、エージェントプログラムが含まれているか。	済みプログラム一覧」に、“System Center Operations Manager 2007 R2 Agent”が含まれていた。
4	運用監視サーバの運用監視コンソールで北海道メールサーバの稼働状態を表示出来るか。	運用監視サーバの運用監視コンソールで北海道メールサーバを認識し、北海道メールサーバの状態が緑色のアイコンで表示された。

京都府データセンターの運用監視サーバから、LGWANを經由して北海道データセンターのメールサーバを運用監視出来ることを実証するため、北海道メールサーバに接続されているNICを無効化し、意図的なネットワーク障害に発生させた。

ネットワーク障害を運用監視サーバの運用監視用コンソールで検知出来るか以下の項目について確認した。確認結果は下表のとおりである。

表 4-86 確認結果

No.	確認項目	実証の結果	状態が変更するまでに要した時間
1	北海道メールサーバのNICを無効化した後、運用監視サーバの運用監視用コンソールのアイコン表示色がグリーンからグレーに変更されたか。	北海道メールサーバのNICを無効化した後、運用監視サーバの運用監視用コンソールのアイコン表示色がグリーンからグレーに変更された。	40秒

#### D. 結果の考察

実証の結果から得られた内容を踏まえ、その考察について記述する。

##### a. データセンターを經由した仮想マシンの運用監視について

京都府データセンターの運用監視サーバから、LGWANを經由して北海道データセンターのメールサーバのNICの無効化（ネットワーク障害）を検知することが出来たことから、LGWANを經由したデータセンター間の仮想マシンに対する運用監視の機能性を確認した。北海道メールサーバのNICを無効化してから運用監視コンソールのアイコン色を変更するまでに要した時間は40秒であり、この時間については、一般的なシステムにとっては運用上問題のない範囲であると考えられ、自治体クラウドにおいて実用的であるといえる。

実運用においては、自治体クラウド基盤上で稼働するサーバー（仮想マシンを含む）は多様なOS（オペレーティングシステム）で構成され、その数も膨大になることが想定されるため、運用監視システムは、複数のOSに対応し、大規模な監視が可能である製品を選定することが望ましい。

b. 運用監視エージェントのリモートインストールについて

運用監視エージェントを北海道メールサーバにリモートインストールする実証は、「コンピュータとデバイスの管理ウィザード」を実行した結果、エラーが発生し、インストールすることが出来なかった。配布元の京都府運用監視サーバ、配布先の北海道メールサーバ双方のイベントログを確認したが、原因を特定するための有用な情報は取得出来なかった。

原因の仮説として、本実証を実施していた時間帯は京都府データセンターから北海道データセンターへ常時リモートデスクトップ（ターミナル）接続を行っていたところであるが、接続後に頻繁に切断される事象が発生していたことから、ネットワークは不安定な状況にあったと考えられ、その結果、エージェントのセットアップファイル（容量：17,219KB）の伝送が正常に終了しなかった可能性が考えられる。今回のように何らかの原因により運用監視エージェントがネットワーク越しにインストール出来ない場合に備え、

- 各データセンターにエージェントのセットアッププログラムを配布しておく
- ローカルインストールを行う場合の操作手順を明確にしておく

などの対策をしておくことが望ましい。

## 4.3 アプリケーション接続実証

自庁舎にある情報システムを利用する場合と比べ、データセンターに集約された自治体クラウドのアプリケーションについては実際に業務の遂行が可能か、アプリケーションの応答時間は実運用に耐えうるか等現場に様々な不安があることも事実である。本実証においては、自治体クラウドのアプリケーションによっても、自庁舎内の場合と同様に業務が遂行できることが示された。

### 4.3.1 基幹系を含むアプリケーション利用実証

#### (1) 実証の概要・目的

現在、市町村で使用されている環境はサーバとクライアントが存在し、市町村内でネットワークが完結している自治体が多数である。近年、市町村にはサーバやアプリケーションを設置せず、データセンターにある環境へ接続してシステムを利用する運用も増えてきた。

当実証実験では、利用者が市町村の端末から広域ネットワークを介してデータセンターの環境へアクセスし、従来通りの業務が行えること、アクセス権限のない利用者による不正アクセス等の防止はできることを確認した。さらに遠く離れたデータセンターへ接続することや広域ネットワークを介することでレスポンスの時間の影響はないのか確認した。

特に今回は、住民情報系、税系、福祉系のサービス利用について紹介する。

#### (2) 住民情報系・税系システム接続実証

##### ア) 実証の内容

###### A. 前提条件

共同利用型アプリケーション（以下、基幹業務支援システム）の住民基本台帳及び個人住民税システム）を使用し、ネットワークを介してデータセンターに接続した際の動作検証を行った。

基幹業務支援システムは既に複数の自治体で稼動しているシステムであるため、システム自体の網羅的な動作検証は不要と判断し、システム自体の動作検証を行うのではなく、クラウド環境においてシステムが問題なく運用できるかについて検証を行った。したがって、検証項目としては、更新系、照会系、帳票系、一括更新処理等の処理の種類ごとに代表的な機能について検証を行った。

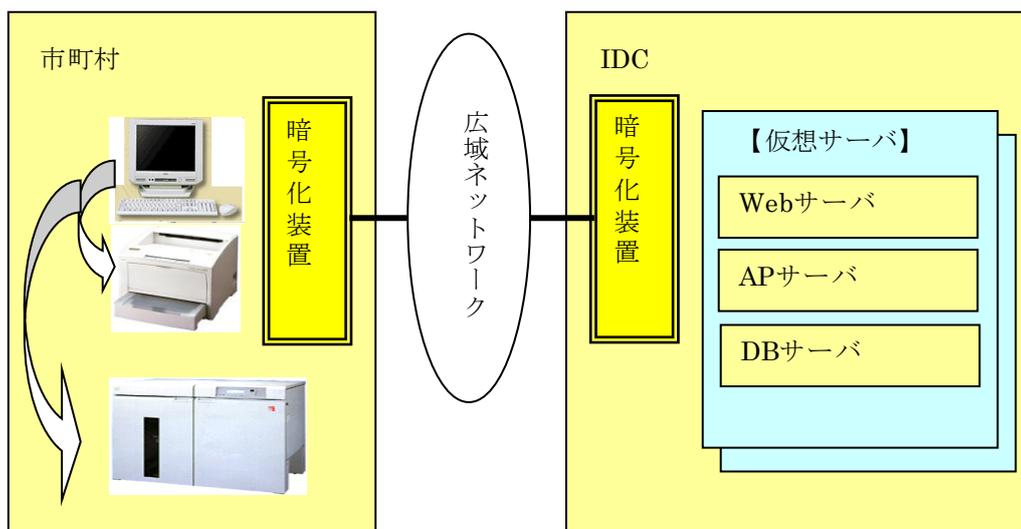


図 4-42 住民情報系・税系システム全体図

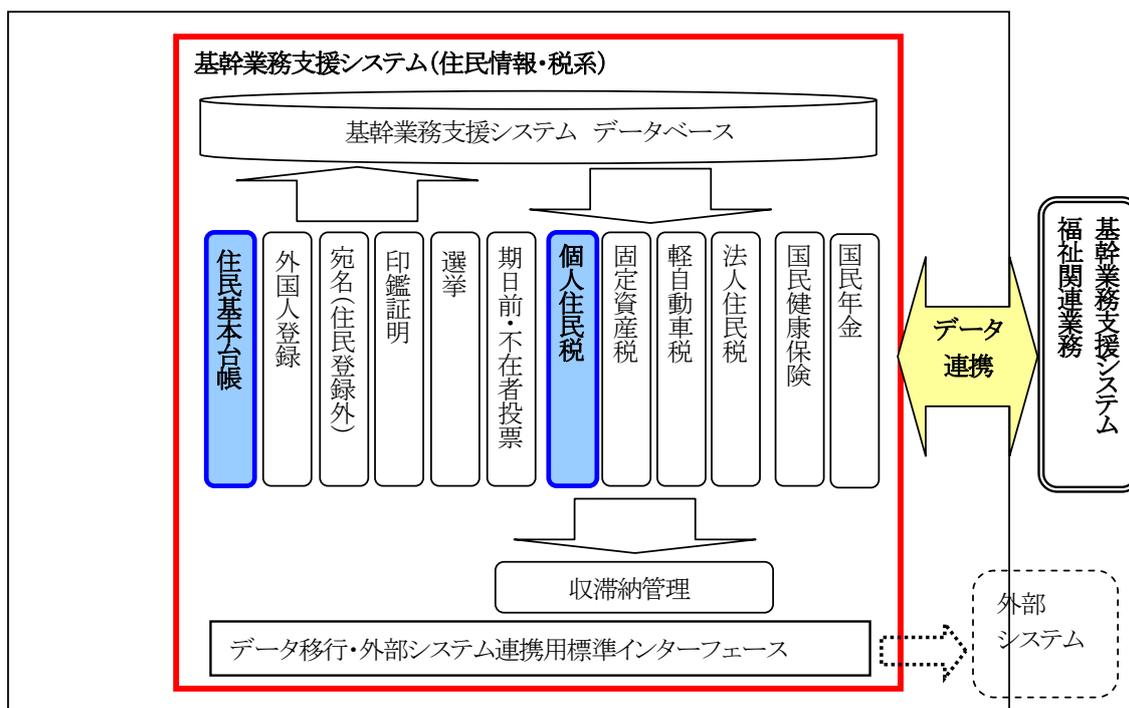


図 4-43 住民情報系・税系システム構成図

B. 実施環境

市町村よりデータセンターへネットワークを介して接続する。ネットワークについては30Mbpsの帯域を利用する。市町村職員が利用するクライアント（運用端末）を用いて実証実験を実施する。

## a. 対象業務

基幹業務支援システム（住民基本台帳システム、個人住民税システム）

## b. 対象データ

基幹業務支援システム導入顧客データを対象とし、実証実験を実施する。

人口 55,093 人

- ・住民基本台帳登録件数（履歴含む） 423,167 件
- ・個人住民税データ件数 325,953 件
  - 平成16年度 43,710 件
  - 平成17年度 43,175 件
  - 平成18年度 43,958 件
  - 平成19年度 48,805 件
  - 平成20年度 46,798 件
  - 平成21年度 51,489 件
  - 平成22年度 48,018 件

## C. 実施手順

## a. 業務の遂行

市町村の端末において自治体職員ユーザとしてシステムにアクセスし、住民基本台帳・住民税の業務処理に係る代表的な操作（異動、照会、帳票出力、一括処理）が問題なくできることを確認する。確認結果は「テスト仕様書兼成績書」に記録する。

## b. 時間計測

市町村の端末において自治体職員ユーザとしてシステムにアクセスし、各操作に要する時間を計測する。また、目標とする基準時間内に処理が完了することを確認することにより、クラウド実証実験の環境において性能面での問題が発生しないことを検証する。

## c. アクセス権

表 4-87 アクセス権設定一覧

職員ID	パスワード	操作権限
9213	****	全業務操作権限付与
9215	****	住民基本台帳操作権限付与
9216	****	個人住民税操作権限付与

1. アクセス権限の無い利用者がアクセスする。
  - (ア) 許可されていないクライアント（運用端末）端末からシステムにアクセスする。
  - (イ) 許可されていない職員IDによりシステムにアクセスする。
2. 利用者ごとの業務権限にて操作する。
  - (ア) 全業務の操作権限を持つ職員IDによりシステムにアクセスする。
  - (イ) 住民基本台帳業務のみ操作権限を持つ職員IDによりシステムにアクセスする。
  - (ウ) 個人住民税業務のみ操作権限を持つ職員IDによりシステムにアクセスする。
3. 操作ログを取得する。

## イ) 実証の結果

### A. 業務の遂行

実証実験環境において、市町村側クライアント（運用端末）から、ネットワークを介して自治体職員ユーザとしてシステムにアクセスし、住民基本台帳システム及び個人住民税システムの稼動検証を実施した。検証対象となる機能において、システムが正常に動作していることを確認した。

表 4-88 業務遂行時のテスト結果

No	テスト項目	結果
1	「転入」の処理が正常に終了するか	OK
2	「転居」の処理が正常に終了するか	OK
3	「転出」の処理が正常に終了するか	OK
4	住民票原本が正常に出力されるか	OK
5	住民票が正常に出力されるか	OK
6	転出証明書が正常に出力されるか	OK
7	記載事項証明書が正常に出力されるか	OK
8	「異動事由別一覧集計表」が正常に出力されるか	OK
9	「集計表」にて「行政区別年齢別集計表」が正常に出力されるか	OK
10	「住民閲覧台帳」が正常に出力されるか	OK
11	受理通知更新確認リストが正常に出力されるか	OK
12	受理通知一括更新処理処理が正常に終了するか	OK
13	強制修正処理が正常に終了するか	OK
14	全国住所辞書一括更新処理が正常に終了するか	OK
15	制限登録処理が正常に終了するか	OK

### B. 時間計測

実証実験環境において、市町村側クライアント（運用端末）から、ネットワークを介して自治体職員ユーザとして基幹業務支援システムにアクセスし、一連の操作に係るレスポンス時間の測定を行なった。実証実験環境に

において、自庁導入方式（サーバ機器を自庁内に設置し、ローカルネットワークにて運用する方式）と同様のレスポンスで動作するが確認できた。

表 4-89 時間計測のテスト結果

No	テスト項目（時間測定）	結果（秒） 1回目/2回目
1	基幹業務支援システムメニューの起動における時間測定	1.24/0.85
2	職員認証（ログイン）画面にて職員コード及びパスワードを入力し、業務メニュー表示の時間を測定する。※入力時間は考慮しない。	1.23/1.08
3	業務メニューより照会システム機能に係る時間を測定する。	0.87/0.69
4	住記メニューから個人照会を選択(エントリー画面が表示されるまで)	0.64/0.50
5	該当者選択画面からメイン画面への遷移(個人コード指定)	1.46/0.90
6	該当者選択画面から該当者一覧画面への遷移 (生年月日、カナ氏名検索)	1.87/0.69
7	該当者選択画面で該当者を選択してから個人照会画面への遷移	1.46/0.91
8	個人照会から世帯照会への遷移（業務連携機能検証）	1.51/1.53
9	個人照会から住民票発行指示への遷移（業務連携機能検証）	2.50/1.69
10	個人照会から記載事項証明発行指示への遷移（業務連携機能検証）	2.70/1.61
11	住民票発行指示画面において、発行ボタン押下後、世帯（4人世帯）の1人分の「住民票写し」が発行されるまで(プレビュー画面表示まで)	4.20/3.98
12	住民票発行指示画面において、発行ボタン押下後、世帯全員（4人世帯）の「住民票写し」が発行されるまで(プレビュー画面表示まで)	4.16/3.98

### C. アクセス権

#### ① アクセス権限の無い利用者がアクセスする。

許可されていないクライアント（運用端末）端末からシステムにアクセスした場合、認証画面にて対象端末の使用が許可されていない旨のメッセージを表示し、システムが使用できないことを確認した。また、許可されていない職員IDによりシステムにアクセスした場合、職員IDに誤りがある旨のメッセージを表示し、システムが使用できないことを確認した。

#### ② 利用者ごとの業務権限にて操作する。

全業務の操作権限を持つ職員ID、住民基本台帳業務のみ操作権限を持つ職員ID及び個人住民税業務のみ操作権限を持つ職員IDにてシステムにアクセスし、認証後にシステムの利用範囲が各職員IDにひもづく操作権限の範囲内に制限されることを確認した。

#### ③ 操作ログを取得する。

基幹業務支援システムを利用し、当実証実験における操作ログを取得す

る。

以上の検証結果より、アクセス権限の無い利用者がアクセスし、正規の利用者以外は利用できない事、また、利用者に付与される操作権限によりシステムの利用範囲が制限できることを確認した。

#### ウ) 実証の考察

共同利用型アプリケーションの接続実証の結果から、各業務システムの動作及び性能面においてクラウド環境での運用に問題ないと考えられる。情報漏洩やデータ改竄に対する対策、ログの追跡に関して、アプリケーションが実装する基本機能が正常に動作することを確認した。また、基幹業務支援システムの導入に際しては、物理的認証機能<sup>12</sup>の調達を前提としており、システム全体としてのセキュリティは一定確保できると考える。

採取したログを分析した結果、認証画面通過後（認証許可後）の操作ログは記録されているが、以下のアクセスが発生した場合の操作ログが記録されないことが判明した。

- ① 許可されていないクライアント（運用端末）端末からシステムにアクセスする。
- ② 許可されていない職員IDによりシステムにアクセスする。

基幹業務支援システム開発時に実施したネットワーク帯域検証において、10Mbps以上の帯域では、若干の違いはあるものの同様のレスポンスを示しているが、10Mbps未満の帯域においては、起動に2倍程度の時間を要するなど、著しくレスポンスが低下することが報告されている。本実証実験において、市町村からデータセンターまでのネットワーク環境は30Mbps程度の帯域を利用したが、10Mbps以下の環境での業務運用は困難となることが予想される。

<sup>12</sup>物理的認証機能とは、ICカードを利用したセキュリティ環境と同等以上の物理認証によるセキュリティ機能

表 4-90 ネットワーク帯域検証の参考結果

業務（処理）		遅延 帯域	20Ms			
			100Mbps	10Mbps	1Mbps	
住民情報システム	起動	1回目（秒）	2.09	1.53	2.28	
		2回目（秒）				
	ログイン	1回目（秒）	1.38	1.84	3.12	
		2回目（秒）				
	照会	該当者選択	1回目（秒）	1.62	1.56	2.75
			2回目（秒）	1.22	1.03	2.44
		個人選択	1回目（秒）	1.63	1.50	2.65
			2回目（秒）	1.68	1.56	2.69
	異動	起動	1回目（秒）	77.34	89.28	190.25
			2回目（秒）	77.97	87.69	189.75
		該当者	1回目（秒）	3.54	4.00	6.97
			2回目（秒）	1.44	1.38	1.65
		個人	1回目（秒）	5.22	5.53	8.56
			2回目（秒）	1.93	1.88	2.94

- ※ 検証結果は別の検証環境において測定された値であり、本実証実験との関連性はない。
- ※ 遠距離間（500Km）でのネットワーク接続を想定し、ネットワーク上に20Ms遅延を発生させた。

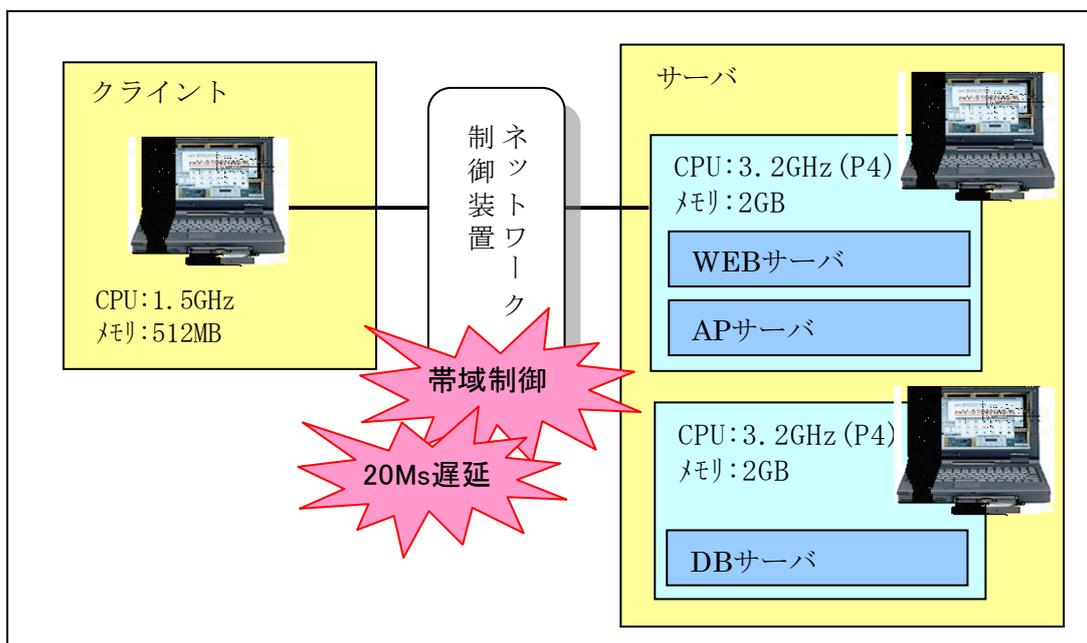


図 4-44 検証環境図

- WEBサーバ、APサーバの機能は同一サーバで稼働させた。

(3) 福祉系システム接続実証

ア) 実証の内容

A. 前提条件

共同利用型アプリケーション（以下、基幹業務支援システム）の福祉医療業務を使用し、システムに接続した際の動作検証を行った。環境については下の図のように自治体からデータセンターのクラウド環境へ接続可能な端末（クライアントPC、プリンタ）で調査を行った。また、基幹業務支援システムは、複数の自治体で既に実稼動しているシステムであることから主な動作のみ確認することで問題ないと判断し、それをもとにテストを行った。さらに、即時帳票やバッチ帳票、画面等の同じ動作をするものについては、代表のものだけを実行し、テストを行った。

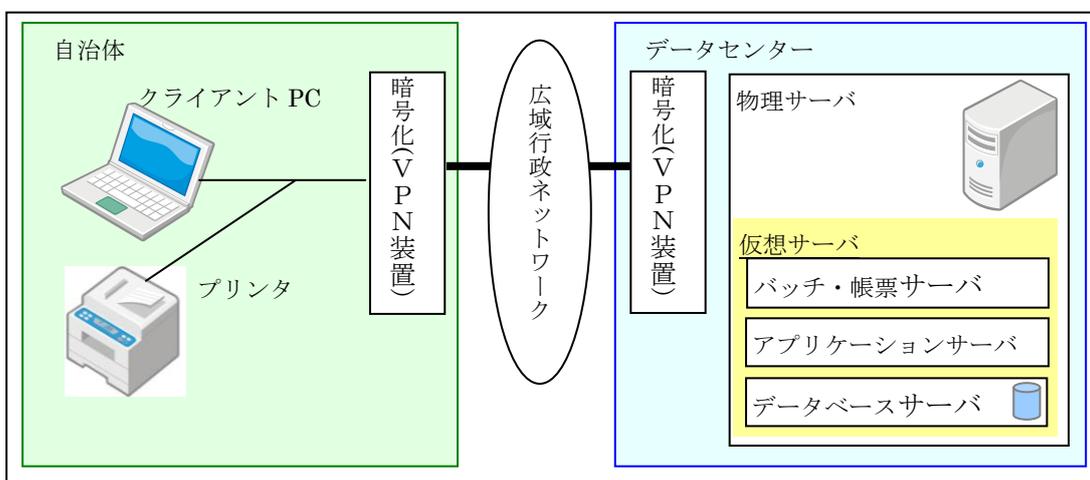


図 4-45 福祉系システム全体図

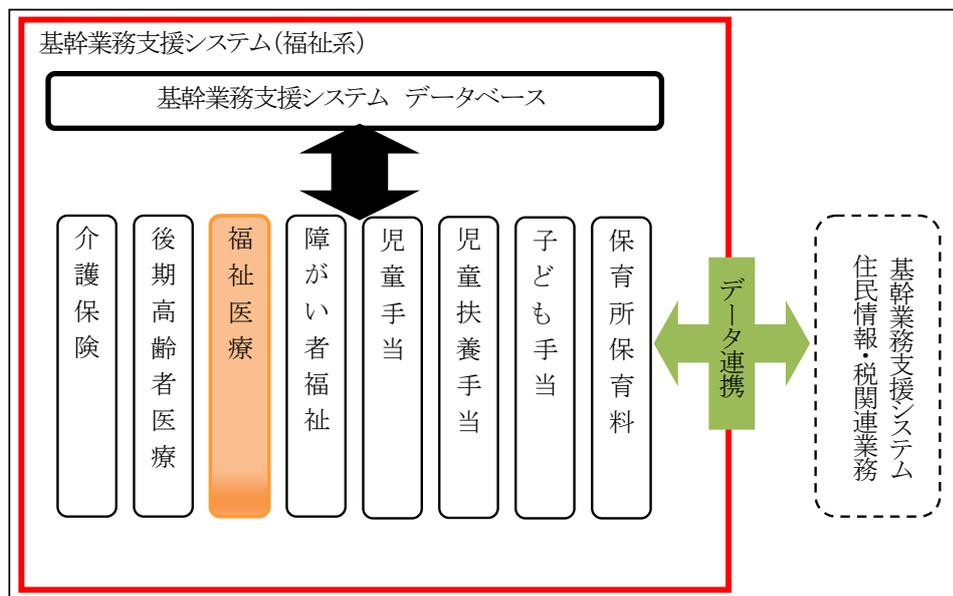


図 4-46 福祉システム構成図

## B. 実施環境

実証実験を実施した環境は以下のとおりである。

### a. 環境

環境としては、自治体のクライアントから広域行政ネットワークを介してデータセンターにあるサーバに接続し実施した。（ネットワーク帯域：30Mbps）

一方、処理速度比較のための環境は、データセンター内に設置してある管理用端末でシステムを実行し、広域行政ネットワークを介さない環境で実施した。（ネットワーク帯域：100Mbps）

### b. 使用データ

約55,000人規模の自治体の住民情報システムデータを使用した。

### c. 使用ハードウェア

実証環境で使用したハードウェアは以下のとおりである。

表 4-91 アプリケーション接続実証使用ハードウェア一覧

No	名称	台数	スペック			
			CPU	メモリ	HDD	OS
1	バッチ・帳票サーバ (仮想マシンとして用意)	1	Intel®Xeon® E5503 2 GHz	4GB	C : 30GB D : 30GB	Windows Server 2008 Standard(64bit)
2	アプリケーションサーバ (仮想マシンとして用意)	3	Intel®Xeon® E5503 2 GHz	4GB	C : 30GB D : 30GB	Windows Server 2008 Standard(64bit)
3	データベースサーバ (仮想マシンとして用意)	1	Intel®Xeon® E5503 2 GHz	4GB	C : 30GB D : 30GB E : 250GB	Windows Server 2008 Standard(64bit)
4	クライアント	1	Intel®Core i5	4GB	C: 160GB	Windows7 Professional(32bit)

表 4-92 処理速度比較用ハードウェア一覧

No	名称	台数	スペック			
			CPU	メモリ	HDD	OS
1	バッチ・帳票サーバ (仮想マシンとして用意)	1	Intel®Xeon® E5503 2 GHz	4GB	C : 30GB D : 30GB	Windows Server 2008 Standard(64bit)
2	アプリケーションサーバ (仮想マシンとして用意)	3	Intel®Xeon® E5503 2 GHz	4GB	C : 30GB D : 30GB	Windows Server 2008 Standard(64bit)
3	データベースサーバ (仮想マシンとして用意)	1	Intel®Xeon® E5503 2 GHz	4GB	C : 30GB D : 30GB E : 250GB	Windows Server 2008 Standard(64bit)
4	クライアント	1	Intel®Core i3	4GB	C : 69GB D : 68GB	Windows7 Professional(32bit)

d. 使用ソフトウェア

実証環境で使用したソフトウェアは以下のとおりである。

表 4-93 アプリケーション接続実証使用ソフトウェア一覧

No	名称	使用用途
1	Windows Internet Explorer 8	システム稼働ブラウザ
2	Adobe Reader 9	帳票出力
3	Microsoft Excel 2003	CSV 出力

## C. 実施手順

自治体端末において自治体職員ユーザとしてデータセンターにあるサーバの基幹業務支援システムにアクセスし、ログインできるか。福祉医療業務処理の帳票プレビュー、画面展開、バッチ処理等が正常に動作するかをテスト項目の手順に従って行った。

また、遠く離れたデータセンターへ接続し、ネットワークを介することによるレスポンスの影響を調べるためにその処理の操作にかかる時間の計測を行った。

### イ) 実証の結果

福祉医療の業務において従来通りの動作が行えるのかを確認するため、自治体の端末からデータセンターの環境へアクセスし、共通テスト、受給者台帳処理テスト、給付台帳処理テスト、ファイル取り込みテストと、大きく4分類のテストを行った。

#### <以下、各テストの説明>

- ① 共通テスト…住民記録、宛名納付業務を使用しての基本的処理の確認
- ② 受給者台帳処理テスト…福祉医療業務特有処理を受給者台帳で確認
- ③ 給付台帳処理テスト…福祉医療業務の給付台帳でのテスト。ただし②と同じ業務のため台帳が参照できるか等の基本動作の確認
- ④ ファイル取り込みテスト…自治体の端末にあるローカルファイルを処理できることを確認

## A. 共通テスト

基幹業務支援システムの総合メニュー・メンテナンス画面にログインし、トップ画面が表示されることが確認できた。次に、台帳検索後に台帳参照を行った。即時・バッチ共に帳票の作成とプレビューがエラーなくできる事が確認できた。上記のことから、クラウド環境においてもログインと基本処理（業務特有の処理以外）がC/S環境と同様に動作することが可能であった。

表 4-94 共通テスト確認事項

No.	内容	検証結果
1	メンテナンス画面にログインできるか	OK
2	システムにログインし、総合メニューが表示されるか	OK
3	即時帳票が正しくプレビューができるか	OK
4	バッチ処理(スプールあり)が正しく実行されるか	OK

## B. 受給者台帳処理テスト

新規で申請登録をした後、登録した台帳の参照を確認した。その後、即時・バッチ共に帳票の作成とプレビューを確認した。リカバリ処理後は、作成した指定年度分の台帳が削除されていることを確認した。上記のことから、クラウド環境においてもログインと基本処理（業務特有の処理以外）がC/S環境と同様に動作することが可能であった。

表 4-95 受給者台帳処理テスト確認内容

No.	内容	検証結果
1	異動事由（分類：認定）を行い、入力した申請内容が登録されるか（No1）	OK
2	バッチ処理の「所得判定処理」を行い、正常に行われるか（No15）	OK
3	「リカバリ処理」を行い、その年度分の台帳が削除されるか（No24）	OK
4	バッチ帳票のプレビューができるか（No17）	OK
5	即時帳票のプレビューができるか（No35）	OK

## C. 給付台帳処理テスト

給付台帳で受給者を検索し、対象の台帳の参照ができることを確認した。また、内容の編集が可能であることを確認するために、金額を変更して保存し、データが更新されていることを確認した。上記のことから、クラウド環境においてもログインと基本処理（業務特有の処理以外）がC/S環境と同様に動作することが可能であった。

表 4-96 給付台帳処理テスト確認内容

No.	内容	検証結果
1	給付台帳の検索画面が表示されること	OK
2	指定された条件から受給者を検索できるか	OK
3	対象の受給者情報が正しく照会できるか	OK
4	異動処理（金額情報が更新）ができるかされるか	OK

## D. ファイル取り込みテスト

バッチサーバからファイルの指定をしてバッチ処理が実行されることを確認でき、クライアントPCにあるローカルドライブからもフルパスを指定してファイル取り込み処理がエラーなく正常に動作することを確認した。上記のことから、クラウド環境においてもログインと基本処理（業務特有の処理以外）がC/S環境と同様に動作することが可能であった。

表 4-97 ファイル取り込みテスト確認内容

No.	内容	検証結果
1	バッチサーバからのファイルの取込が可能であるか	OK

2	クライアント PC にあるローカルドライブにアクセスできるか	OK
3	クライアント PC にあるローカルドライブにフルパスで指定できるか	OK

また、アクセス権限について、権限のないユーザ・端末がログインできないことを確認するために権限のないクライアント端末と職員IDで基幹業務支援システムへアクセスし、認証画面にて対象端末の使用許可がされていない旨のメッセージが表示され、システムが使用できないことを確認した。次に、権限のあるユーザでのログイン確認をするために次のような職員IDを用いて確認を行い、認証後にシステムの利用が操作権限の範囲内に制限されることを確認した。

- ◆全業務の操作権限のある職員ID
- ◆福祉医療業務のみに操作権限のある職員ID
- ◆総合システムにのみ操作権限のある職員ID

アクセス権限と関連して、今回の実証実験中のログを取得し、ログインの成功時と失敗時の記録内容を確認した。ログには、下図のような項目と内容が記録されており、ログインが成功すれば、全項目が埋まるようになっている。失敗した場合、ログイン失敗が記録され、どの端末からいつアクセスがされたかがわかり、不正アクセス内容が把握できるようになっている。

表 4-98 アクセスログファイルの内容例

操作日時	操作者ID	操作者名	端末名	操作ID	操作名	メンテナンス
2011/1/1 12:34	Admin	管理者	XXXX999	LOGONSUCCESS	ログイン(成功)	○

※「メンテナンス」項目が空白の場合は総合メニューへのログインを表し、○はメンテナンス画面へのログインがあったことを示す。

次に、自治体側のクライアントPCから、ネットワークを介して基幹業務支援システムにアクセスし、ログインから基本処理の一連の操作のレスポンスタイムの計測を行った。ネットワークを介してデータセンターに接続する「クラウド環境」と、ネットワークを介さずデータセンターでの環境で実行した場合の「データセンター内環境」において、それぞれのレスポンスタイムを比較した。時間計測実施内容、及びそれぞれの使用データ件数は以下のとおりである。データ件数は同じデータベースを使用するため同じ件数となる。

表 4-99 時間計測実施内容一覧

No.	操作	手順
1	ログイン画面表示	トップページの [実行] ボタンを押す。
2	総合メニュー表示	パスワードを入力し [OK] ボタンを押す。

3	検索画面表示	カナ氏名を全件検索し [検索開始] ボタンを押す。
4	台帳表示	住民を選択し [選択] ボタン押す。
5	即時印刷画面表示	[印刷] - [住民票写し(個人票)] を選択し、 [発行] ボタンを押す。
6	バッチ処理実行	[随時処理] - [個人一覧作成処理] を起動し、 [実行] ボタンを押す。
7	ヘルプファイル表示	[ヘルプ] - [運用操作マニュアル] を選択する

表 4-100 時間計測実施のデータ件数 (福祉システム)

年度	データ件数処理時間	
	クラウド環境 (広域行政ネットワーク使用)	データセンター内環境
平成 16 年度	2477 件	2477 件
平成 17 年度	2475 件	2475 件
平成 18 年度	3466 件	3466 件
平成 19 年度	3531 件	3531 件
平成 20 年度	3232 件	3232 件
平成 21 年度	3336 件	3336 件
平成 22 年度	3198 件	3198 件

- 人口規模：55,000人の自治体
- データ内容：福祉医療の受給者台帳の件数

レスポンスタイムは下図のような結果となった。グレーで反転している箇所のように、かなりの数値差やばらつきがみられるが、これらは初回実行時に多く見られることから、接続の処理やネットワークの使用状況などの環境によるものと考えられる。2回目、3回目のデータを有効数値として考え、使用しているクライアントのスペックも異なることを考慮すれば、広域行政ネットワークを介するクラウド環境であってもレスポンスタイムとしては影響がないと考えられる。

表 4-101 時間計測結果

No.	内容	表示 件数 (件)	処理時間 (秒)					
			クラウド環境 (広域行政ネットワーク使用)			データセンター内環境		
			初回	2回目	3回目	初回	2回目	3回目
1	ログイン画面 表示		15.92	04.70	04.17	22.33	12.22	02.92
2	総合メニュー 表示		08.73	07.76	08.29	08.21	05.78	06.18
3	検索画面表示		10.37	05.34	05.14	07.50	06.58	06.32
4	台帳表示		04.38	04.60	03.82	08.22	03.59	03.12
5	即時印刷画面		01.31	00.77	00.95	01.67	00.80	00.87

	表示							
6	バッチ処理実行	1,335	04.71	07.41	05.14	20.00	09.42	07.42
7	ヘルプファイル参照		03.77	02.67	02.80	03.18	01.73	02.93

(単位：秒)

- システムの特性上、必要なモジュールが初回にダウンロードされる仕組みのため、ログイン画面表示の初回処理時間は2回目に比べ遅くなることは事前に分かっている。

#### ウ) 実証の考察

今回の結果から、システムへの接続時に認証が正常に行われ、システムの動作も正常で有ることが確認できた。また、セキュリティ面で情報漏洩やデータ改竄に対する対策、ログの追跡に関して、アプリケーションが想定する基本機能において正常に動作することも確認できた。

また、運用面から見てみると、下図は福祉システムを使用する上での運用の流れについて概要レベルで表したものだが、クラウド環境においても特別な処理が発生するということはなく、自治体側でローカルネットワークの環境で使用している場合と同様の流れになることも当実証実験の中で確認することができた。よって、システム面・運用面ともに利用者側はクラウド環境ということ意識せずにシステムを使用することが可能である。

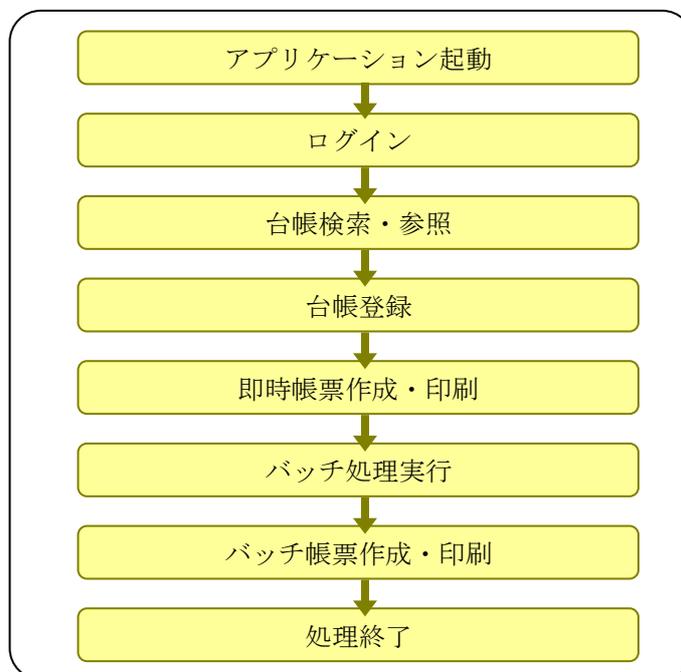


図 4-47 福祉系システムの基本的な操作の流れ

一方でネットワークに関しては、自治体とデータセンター間はVPN装置

によりネットワークレベルで暗号化がされており、IPネットワークを利用して専用線をつないだかのような仮想的な接続が可能であるためセキュリティも十分確保できると思われる。

また、自治体内でネットワークが完結している環境においても言えることであるが、他のネットワークトラフィックや同時アクセスにより、想定外のネットワークトラフィックが発生することが考えられる。ネットワーク速度がボトルネックとなり、レスポンスの低下を招くことも考えられるため、サーバとクライアント間のネットワーク全体についても考慮する必要がある。当実証実験では、30Mbpsのネットワーク帯域で運用に支障なく処理できることが確認できたが、ネットワーク環境によってアプリケーションのレスポンス性能等が低下しないよう、十分余裕を持ったネットワーク環境の構築が望ましい。

なお、あくまで参考ではあるが、当実証実験で使用した福祉系のシステムがネットワークの帯域によってどれだけレスポンスに影響を及ぼすのか検証した。広域行政ネットワークではなく、ローカルネットワーク環境において、100Mbps、10Mbps、1Mbpsの各帯域のレスポンス比較を行った。

環境については、下図のとおりローカルネットワークを構築し、サーバとクライアントとHUBを設置した。ネットワーク回線速度はネットワーク帯域制御ツール（ソフト）を使用して、アプリケーションサーバのネットワークインタフェースカードに対して、100Mbps、10Mbps、1Mbpsに帯域を制御した。

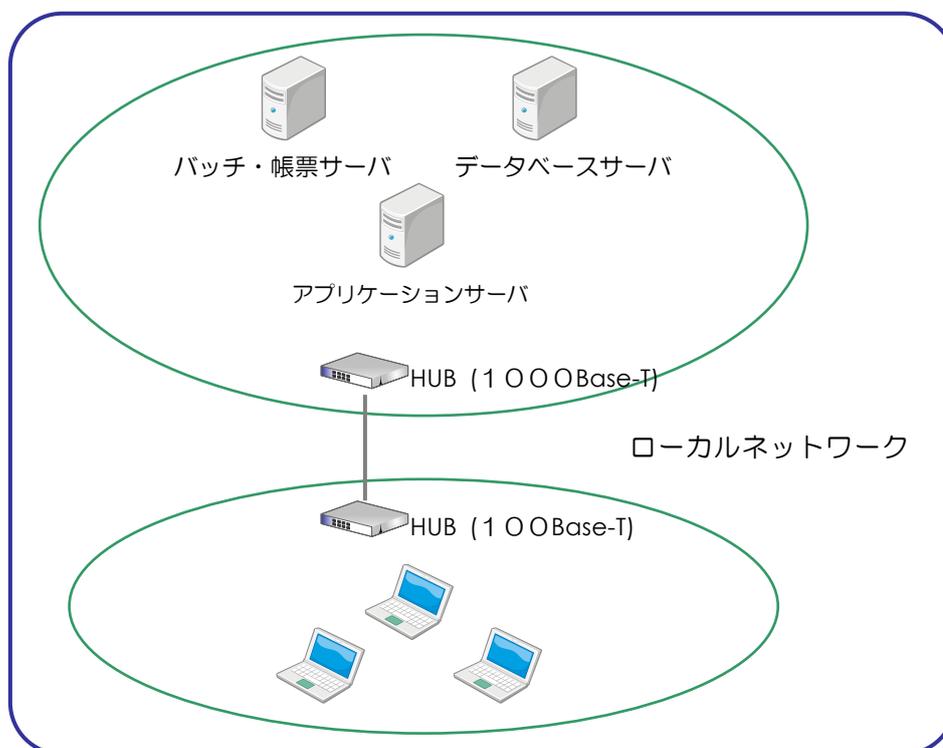


図 4-48 帯域制限を行うローカルネットワーク環境イメージ

実施手順は下表のとおり、システムの代表的な処理を各帯域で実施した。

表 4-102 帯域ごとの処理時間

No.	内容	100Mbps (秒)	10Mbps (秒)	1Mbps(秒)
1	ログイン画面表示	3	3	3
2	総合メニュー表示	7	10	40
3	台帳検索	1	1	4
4	台帳表示	1	2	2
5	即時印刷画面表示	2	2	2
6	ヘルプファイル参照	1	13	105

結果は、ほとんどの帯域でレスポンスにはそれほど差はないが、1Mbpsにおいてはログインやスプール帳票の閲覧のレスポンスに時間を要している。参考値ではあるが、帯域が非常に狭いネットワーク環境では運用が難しいことが言える。そのような環境の中にある自治体においては、自治体にサーバを置き、ローカルネットワークの環境で構築せざるを得ない場面もまだまだあるといえる。

最後に今回は実証できなかったが、複数自治体の利用はどのようになるかについて、環境やシステムの仕組みから検討してみた。下図は、物理的サーバは同じものを使用して複数自治体に対応する場合で、仮想化サーバのネットワークインタフェースカード（以下、仮想NIC）によって自治体を分離する方法である。物理サーバ内には複数の仮想NICがあり、ある仮想NICに接続すればA市へ、別の仮想NICだとB市へつながる、というように設定ができる。また、タグVLANにより仮想化サーバ内の仮想スイッチにてA市とB市の通信データを分離することができ、セキュリティ面も確保できる。これらのことからクラウド環境で複数自治体の稼働も十分可能であるといえる。これは物理サーバを複数自治体で共同利用することにつながり、将来的に多くの自治体で活用する際の参考となる結果だと考えられるため、省コスト省スペース化を行う今後のクラウド環境への期待と希望が見込めるといえる。

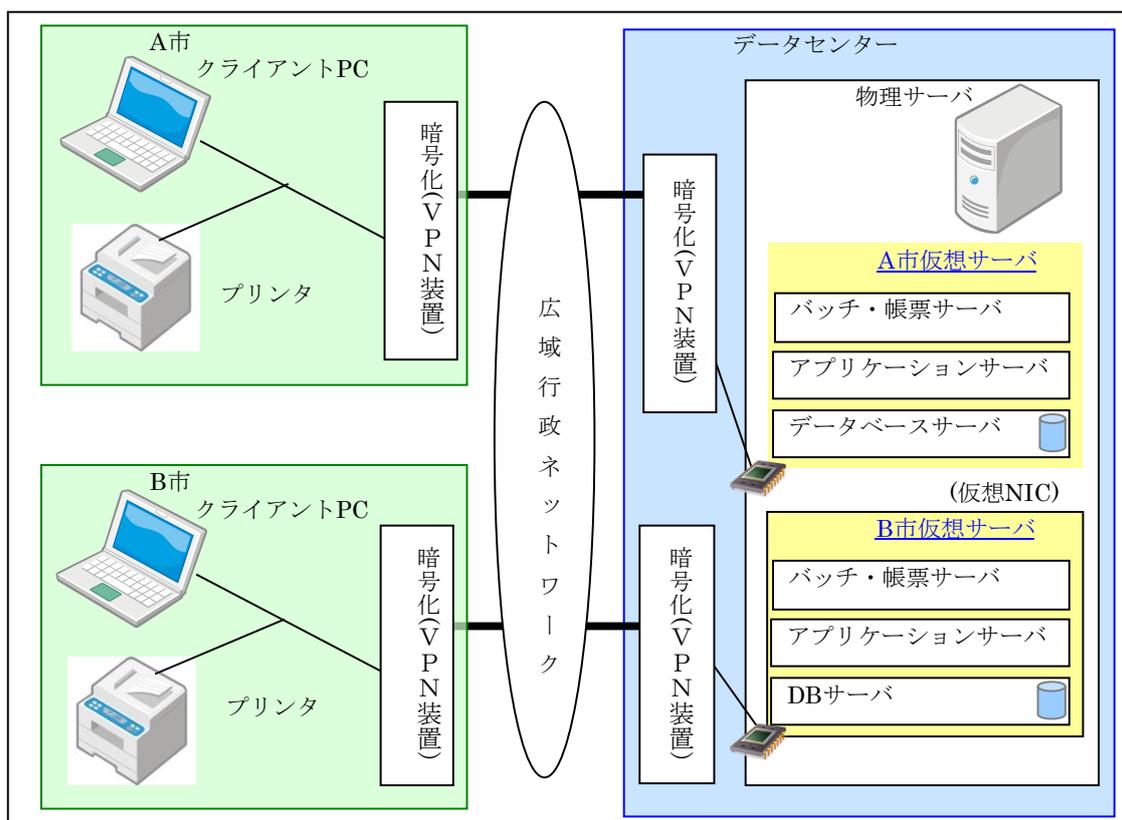


図 4-49 複数団体稼働時のシステムイメージ

### 4.3.2 新規自治体追加実証

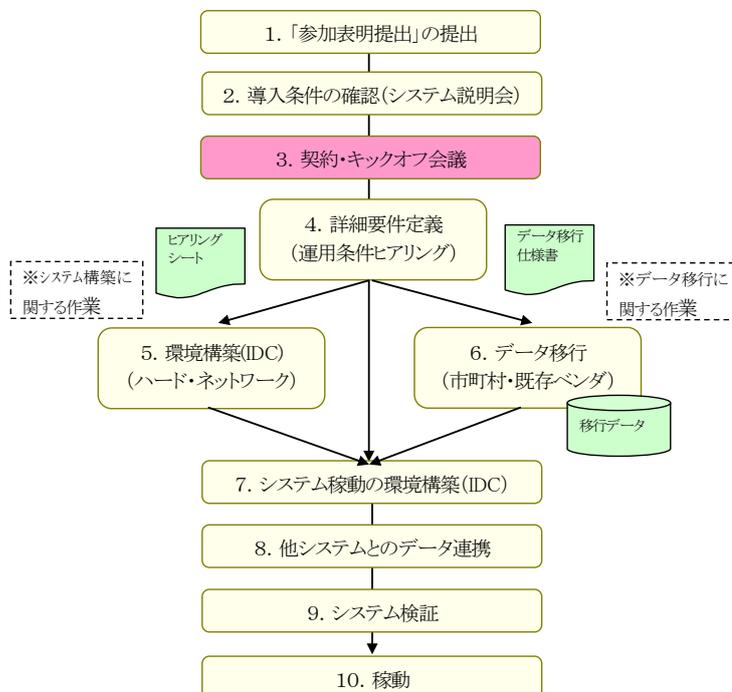
自治体クラウドへの参加団体が増えた場合の追加作業について確認し、これが容易に行えることを手続面、情報システム構築の面の2つの視点から確認した。

#### (1) 実証の概要・目的

京都府下においては平成20年4月より共同利用型アプリケーション（基幹業務支援システム）によるサービス提供を開始しているが、今般、京都府向日市より平成23年4月稼働に向けての参加表明があり、実際にシステム導入作業を通して新規自治体の参加時における一連の手続き検証し、既に稼働中の市町村向に影響なく新規自治体の導入が可能である事を確認した。

#### (2) 実証の内容

京都府下において、基幹業務支援システムは5団体の導入実績があり、システム導入における手続きは、一定、確立している状況にある。当実証実験においては、クラウド環境においても同様の手続きにより、サービス利用が可能となることを検証する。



※ 5. 環境構築、6. データ移行は並行して作業を行う。

図 4-50 新規団体参加のフロー図

既に、京都府自治体情報化推進協議会から新規参加自治体へ「市町村基幹業務支援システムの導入及び運用に係る標準負担金表」（【参考資料1】京都府

参考資料「市町村基幹業務支援システムの導入及び運用に係る標準負担金表」を参照のこと。)が配布されている状況にある。

## ア) 実施手順

以下の手順のより、新規自治体の参加実証を実施した。

### A. 「参加表明書」の提出

新規参加自治体より「参加表明書」(【参考資料1】京都府参考資料「参加表明様式」を参照のこと。)を京都府自治体情報化推進協議会へ提出し、京都府自治体情報化推進協議会が提供する市町村基幹業務支援システムについて、参加意思を表明する。「参加表明書」による確認事項は以下のとおりである。

#### a. システム利用方式

- ① A方式  
標準的な導入運用方式
- ② B方式  
人口規模が小さく、対象業務に精通した市町村担当職員が本システムを理解し、導入・運用作業を市町村が主体となって行うことを前提とする。

#### b. 導入方式・運用方式

- ① ASP方式  
データセンターへネットワークを介して接続する。
- ② 自庁方式  
市町村が自庁内に必要な機器を独自に調達し、基幹業務支援システムを導入する。

#### c. 導入対象サブシステム

導入サブシステムの選択及び導入希望年月(予定)

### B. 導入条件の確認

#### a. 事前説明会の実施

京都府自治体情報化推進協議会から、新規参加自治体向けに導入に係る作業の進め方について説明を行ない、市町村との認識をあわせる。

「市町村基幹業務支援システム導入プロジェクトの今後の進め方について」（【参考資料1】京都府参考資料「『基幹業務支援システム導入プロジェクト』の今後の進め方について」を参照のこと。）

「市町村基幹業務支援システムの導入及び運用に係る標準負担金表」（【参考資料1】京都府参考資料「市町村基幹業務支援システムの導入及び運用に係る標準負担金表」を参照のこと。）

「導入マスタスケジュール雛型（案）」（【参考資料1】京都府参考資料「基幹業務支援システム 導入スケジュール（雛型）案」を参照のこと。）

「実施計画書雛型（案）」（【参考資料1】京都府参考資料「市町村基幹業務支援システム導入プロジェクト実施計画書」を参照のこと。）

### 京都府市町村基幹業務支援システムにおける導入基本方針の確認

#### 【導入基本方針】

京都府自治体情報化推進協議会で定めた市町村基幹業務支援システムの仕様に準じて導入を行う。（現行システムの機能や仕様との差異分析は行わない。）

標準システムが現行業務と適合しない場合、組織体制や条例等の見直しを視野に入れて、できる限り運用面での解決を検討する。

- ① 市町村、協議会、事業者の作業内容と役割分担の確認
- ② プロジェクト体制確保の依頼（プロジェクト責任者、部門責任者、業務担当者）

#### b. 現状分析（確認）

システム導入の具体的検討を実施するため、新規参加自治体における現在のICT状況（資産及び設備の活用）を確認する。

#### c. 詳細確認（システム説明会の実施）

デモシステム及びドキュメントを用いて基幹業務支援システムの機能を説明し、システムを運用する職員の理解を深める。（基幹業務支援システム導入後の業務運用を検証できるレベルの理解度を目標とする。）

基幹業務支援システムを運用する上での課題・問題点を抽出し、対応策を検討する。

#### d. 導入条件の確定

- ① 費用の決定
- ② 導入スケジュールの確定
- ③ 契約内容（契約形態・契約時期）の確定
- ④ 導入に係る諸条件の合意形成

(導入条件を明記した契約案、見積書、仕様書等)

### C. 契約・キックオフ（新規参加団体のシステム導入最終意思決定）

上記、「B.d 導入条件の確定」をもって契約締結を行い、市町村の主催によりキックオフ会議を実施し、作業着手を宣言する。キックオフ会議は、「実施計画書」に基づき主要なプロジェクトメンバーの出席のもと実施し、プロジェクト推進ルール（推進方針、判断基準、情報共有手段等）の説明を行い、プロジェクトメンバーの意思統一を行なう。

### D. 詳細要件定義

#### a. 業務要件の定義（パラメータ設定シートの作成）

各業務でヒアリングシートを用いて、システムが動作するための資源、業務運用条件の確認を行なう。結果は「5.環境構築」作業に反映する。

#### b. データ移行の要件定義

市町村、既存事業者（データ移行業者）に対してデータ移行仕様の説明を行い、データ移行に関する課題・問題点を抽出し、対応策を検討する。

### E. 環境構築

#### a. ハードウェア・ミドルウェアの調達

①ASP方式の場合・・・事業者で調達

②自庁方式の場合・・・市町村で調達

※ クライアント関連機器及びクライアントの調達はASP方式・自庁方式共に市町村で調達する。

#### b. ネットワーク構築

①ASP方式の場合

データセンター内は事業者が構築し、庁舎内は市町村で構築する。

②自庁方式の場合

市町村が構築する。

#### c. OS、ミドルウェア類の設定

要件定義にもとづき、サーバ側の稼動環境の構築を行う。

- 仮想サーバ設計（物理資源の割り当て）
- 仮想サーバ構築

## F. データ移行

基幹業務支援システムの「ファイル項目移行仕様」に従い、既存システムから基幹業務支援システムで取り込み可能なデータが抽出されていることを前提とする。「ファイル項目移行仕様」及び「抽出データのチェックツール」は事業者より新規参加自治体に提供する。

- 「ファイル項目移行仕様」に基づき、データ抽出プログラムを作成する。（市町村、既存事業者）
- 既存システムより移行データを抽出する。（市町村、既存事業者）
- チェックツールを利用し、抽出データのチェックを行う。（市町村、既存事業者）
- 市町村から提供される抽出データを基幹業務支援システムに搭載する。（事業者）
- 基幹業務支援システムに搭載されたデータの論理チェックを行う。（事業者）
- 基幹業務支援システムに搭載されたデータの検証を行う。（市町村）

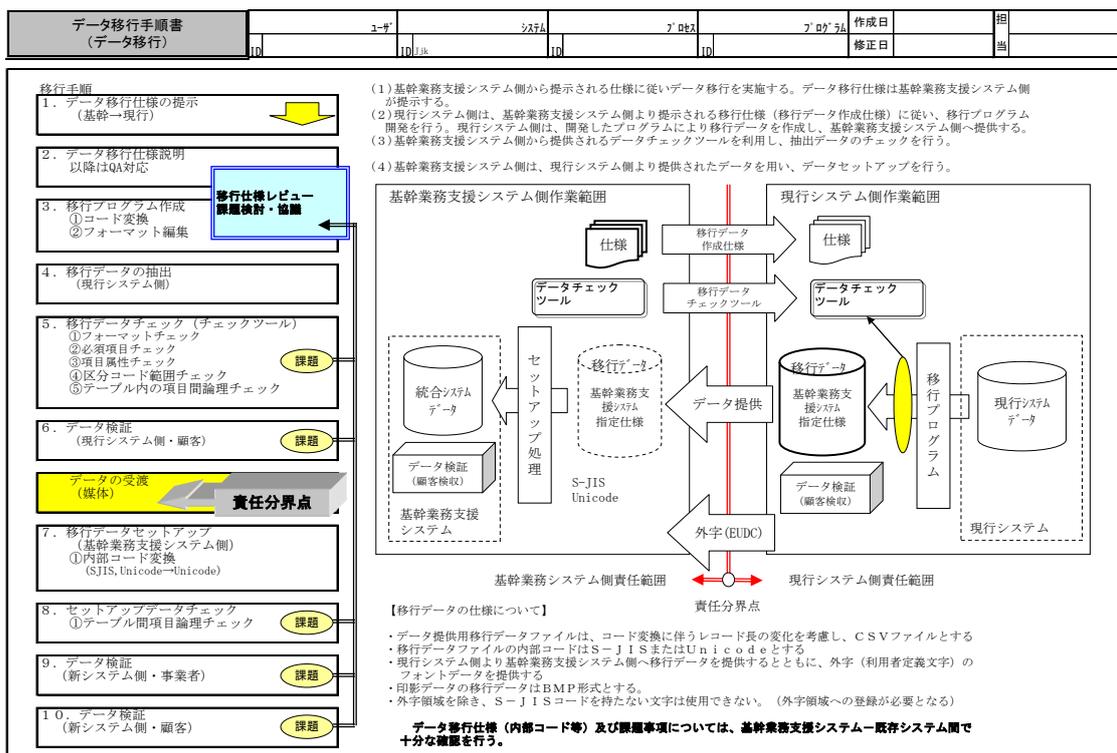


図 4-51 データ移行手順

## G. システム稼動環境の構築 (基幹業務支援システムの実装作業)

- 基幹業務支援システムのインストール
- 動作環境設定

- 業務パラメータ設定
- 動作確認
- ※ クライアント機器の設定、動作確認等は「環境設定手順書」を基に市町村が行う。

#### H. 他システムとのデータ連携

基幹業務支援システムは連携用データベースの仕様を公開している。各システムとのデータ連携については、連携用データベースを利用し、市町村にて対応することとする。

##### a. 既存システムの洗い出し

- 導入市町村で従前から稼働しているシステムを抽出
- 現状のシステム連携方法の確認

##### b. システム連携の方法の検討

- 基幹業務支援システムの要件確定後、関連システムに対するデータ連携の要件定義、仕様確定を行う。
- 仕様確定後、システム連携システム検証までにシステム連携テストを完了する。

#### I. システム検証

- システム検証に向け、「職員研修実施計画書」に操作研修を実施する。
- システム検証は市町村主体で実施する。
  - ◆ システム動作確認
  - ◆ データ検証
  - ◆ 業務運用確認

#### J. 本稼動

システム検証完了後、本稼動用データ凍結及びデータ移行作業を実施する。

データ凍結後の追いかけ入力及び並行処理を実施し、本稼動とする。

#### (3) 実証の結果

上記の手続きにしたがい、新規参加自治体の導入作業を実施した。

- ア) 新規参加自治体より「参加表明書」が京都府自治体情報化推進協議会へ提出

参加表明書	
<p>京都府自治体情報化推進協議会が提供する市町村基幹業務支援システムについて、「市町村基幹業務システムの導入及び運用に係る標準負担金表」を熟知したうえで参加意志を表明しますので、導入に向けた協議及び手続きに着手いただきますようお願いいたします。</p> <p>契約については、予算が確保でき次第、契約させていただきます。</p> <p>なお、市町村基幹業務支援システムにおいて選択する導入方式等及び各サブシステムは、別紙「導入方式等及び導入対象サブシステム」のとおりです。</p>	
1. 導入方式	
(1) 導入・運用方式	<u>A方式</u>
	※市町村、協議会、事業者の3者連携により事業を推進する。(B方式の場合は市町村主体で事業を推進する)
	※システム導入に係る進捗管理、課題管理・検討は事業者が主体となって実施する。
(2) システム利用方式	<u>ASP方式</u>
	※データセンターへネットワークを介して接続
2. 本稼動日	
	平成23年4月

図 4-52 参加表明書 (サンプル)

#### イ) 導入条件の確認

契約までのプロセスにおいて、導入作業着手後の誤解や要件漏れによる課題発生を避けるため、契約締結までの現状分析及び詳細確認（システム説明会）に時間を要した。この工程においては、新規参加自治体の業務担当者が基幹業務支援システムの機能を十分に理解し、導入後の業務運用を定義しつつ、実業務とのギャップ（課題）を抽出できることが重要である。抽出された課題への対応を含めて事業全体の概算費用が確定したが、費用の確定までに予想以上の時間を要する結果となった。また、仮想化サーバにおける稼動実績がなく、機器構成の決定が遅れたため、キックオフ会議前後で導入スケジュールを見直す必要が発生した。

#### ウ) キックオフ会議

事業全体の費用軽減のため、導入基本方針に則り可能な限り個別改修は実施せず、基幹業務支援システムの標準機能を利用することを前提にシステム導入作業に着手した。

導入基本方針については、キックオフ会議にて新規参加自治体の業務担当部門に通知され、関係者の意志統一が行なわれた。

#### エ) 詳細要件定義

導入作業中の連絡については連絡票を活用し、口頭のみやりとりをさけて記録をのこすこととした。また、発生した課題についてはすべて検討課題

台帳に記録し、新規参加自治体と情報を共有しつつ、課題解決に向けての協議を行った。

市町村名		固定システム		版	作成日	作成者	頁					
〇〇市				初版			4					
				改版			11					
I【状況】未着・仕掛・完了 【分類】質問・検討・確認												
No.	発生日	状況	発生元	分類	業務	検討事項・問題点	原因・理由	回答・対応・対策	期限	対応者	完了日	参考資料
1	H22.5.28	仕掛	〇〇市	要望	固定	本市では、評価(公課)証明には構造、用途を出力していません。課税明細書には登録の出力をせずに、別添の通りの主体構造の出力をしてほしい。		基幹業務支援システムでは該当する機能がありません。 〇〇市様の運用状況等をお聞きして、検討することになります。 主体構造の表示については、画面表示用、リスト出力用、課税明細書・証明書出力用と表示内容をパラメータ設定により変更が可能です。 2010081KKの証明書の出力内容を空にしておけば構造は印字されません。ただし、証明書の発出し欄はそのまま印字されます。また、証明書と課税明細書は同一の設定を参照しているため、出力内容を空に設定されるのであれば、証明書下の文言で家庭構造は印字されません等の文言を設定することをご検討ください。				連絡票管理№(〇〇-固定-1から4)
2	H22.5.28	仕掛	〇〇市	要望	固定	各客帳には登録欄にコードで出力してほしい。		基幹業務支援システムでは該当する機能がありません。 〇〇市様の運用状況等をお聞きして、検討することになります。 2010081KKの名義帳の表示用文言をコードで設定すればコード表示はできますが、他の外部帳票もコードで印字されます。印字内容についてご確認ください。				連絡票管理№(〇〇-固定-1から4)
3	H22.5.28	仕掛	〇〇市	要望	固定	評価調書を作成してほしい。		基幹業務支援システムでは該当する機能がありません。 〇〇市様の運用状況等をお聞きして、検討することになります。 2010081KKの基幹業務支援システムから自動画面で確認できる項目のほかにCSVで出力することができるので、〇〇市様で加工していただけないでしょうか。家庭データ抽出処理画面イメージと家庭CSVファイルのサンプルを提示いたします。				連絡票管理№(〇〇-固定-1から4)
4	H22.5.28	仕掛	〇〇市	要望	固定	家庭の登記床面積については、評価(公課)証明及び名義帳には出力しないでほしい。		基幹業務支援システムでは該当する機能がありません。 〇〇市様の運用状況等をお聞きして、検討することになります。 2010081KKのゼロであれば*が印字されますが、ゼロ以外であればそのまま印字されます。 zeroes 移行時に登記床面積をゼロで移行することもできますが、登記床面積が揃っていないものだけを特定して登記床面積をゼロにすることは難しいと思われます。				連絡票管理№(〇〇-固定-1から4)
5	H22.5.28	仕掛	〇〇市	要望	固定	多用途の欄の「住宅以外」の明細はどのように表示することが可能か。		用途毎に課税分割を行うことで、用途毎の明細を出力する対応が可能です。 2010081KKの用途ごとの床面積、評価額を管理することができないため、現行システムで管理されている用途ごとの評価額などの内容を備考欄に移行する等の対応が必要ですが、また、備考欄に移行した場合は、用途ごとの評価額などは手入力による管理となります。				

図 4-53 検討課題台帳（課題管理）固定資産税

システム導入の基本方針としては、「個別改修は行なわず、標準パッケージの標準機能で運用する」ことであったが、運用による課題解決ができず以下の要件について、個別改修を実施することとなった。個別改修については、標準負担金以外に開発費及び保守費用の追加が発生し、新規参加自治体が負担することとなった。

導入作業中に発生した改修候補は以下のとおりである。

表 4-103 カスタマイズ候補一覧（サンプル）

No.	業務	関連資料	見積要件	依頼詳細
1	固定資産税	連絡票 (〇〇-固定-54)	納税通知書と課税明細書のカスタマイズについて	基幹業務支援システムのパッケージの納税通知書・課税明細書では、前システムで作成していた従来の納税通知書・課税明細書よりも、封入作業、郵送作業ともに時間がかかり、それに伴う人件費、郵送費の増大が見込まれることから、納税通知書・課税明細書のカスタマイズを要望します。
2	固定資産税	連絡票 (〇〇-固定-55)	償却資産申告書・明細書のカスタマイズについて	基幹業務支援システムのパッケージの償却資産申告書・明細書は3連の用紙にドットプリンターを用いて印刷する方式が採用されています。既存システムでは申告書・明細書はオーバーレイ方式となっています。そこで、基幹業務支援システムのパッケージでは、新たにドットプリンターの購入費・維持費、申告書・明細書用紙購入費がかかることから、申告書・明細書をオーバーレイ方式で出力できるようにカスタマイズを要望します。
3	固定資産税	連絡票 (〇〇-収納-70)	固定資産税納税通知書兼納付書兼課税明細書を既存様式とする。	固定資産税業務で現在使用している納付書は、1期から4期までの期別納付書の前に一括納付分の納付書を使用（印刷）しています。（別添資料参照）基幹業務支援システムでは、現システムと同様に一括納付分の納付書を作成することは可能ですか。また、納付書の文言やレイアウトの変更は少しは可能ですか。
4	固定資産税	連絡票 (〇〇-固定-44)	経年減点補正率の適用区分について	家屋の経年減点補正率についてですが、〇〇市では添付の資料のとおり、『課税の構造』欄と『評価用途』欄によって経年減点補正率を算出しております。『用途』欄は課税明細書・名寄帳・縦覧帳簿の出力の際に表示しており、評価額には影響がありません。基幹システムにおいても、『評価用途』に対応する入力項目があるのか教えてください。無ければ、『評価用途』欄に対応する入力項目の新設をお願いします。
5	固定資産税	連絡票 (〇〇-固定-56)	課税明細書に建築年が表示されないことについて	前システムで出力している課税明細書には『建築年』を表示していますが、基幹業務支援システムのパッケージの課税明細書では、『建築年』が表示されません。『建築年』は軽減措置切れの説明にも有用である他、多数の家屋を所有している所有者にとっては、物件を特定する貴重な情報であり、評価額の比較にも必要です。基幹業務支援システムで出力される課税明細書にも『建築年』を表示できるようにカスタマイズを要望します。
6	選挙		選挙名簿の氏名に「ふりがな」をいれる。	
7	選挙		選挙人名簿住所欄拡張	名前の欄と備考欄は狭めずに対応して欲しい。現在住基最高は41文字なので、〇〇市を省略し40文字へ拡張？他に良い方法があれば提案して欲しい。
8	住基		異動事由別人口統計表	月次で行政区ごとの男女別増減内訳（自然増減・社会増減）が必要。
9	収納		税普徴、税特徴、国保の3督促状について	税普徴、税特徴、国保の3督促状について、督促状を圧着化し、未納額のお知らせ欄（今年度の未納額を表示させる。）を入れる。
10	収納		軽自の口振領収書、民税・固定の口振領収書の圧着化	
11	収納		旧の済通で新システムに消しこめるようなOCRのPG	

協議の結果、最終的に確定したカスタマイズ要件は以下のとおりとなった。

- ① 固定資産税業務
  - 証明書に関する印字項目の改修対応
  - 名寄帳、家屋縦覧帳簿に関する印字項目の改修対応
  - 納税通知書兼課税明細書の作成（制定用紙）
  - 償却資産申告書のレーザプリンタ化
- ② 住民基本台帳
  - 行政区ごとの男女別増減内訳（自然増減・社会増減）の作成

オ) 環境構築

機器、OS構成は仮想化サーバ構成とした。また、ヒアリングシートに則り、データベース容量（ディスク資源）等の設定を行なった。

今回の自治体クラウド実証実験の実施に伴い、IDCに新たにサーバを導入した。システム動作確認及びシステム検証作業を通して、仮想サーバ環境において市町村における基幹業務の運用が可能であることが確認できた。

今後発生する新規参加団体の追加については、新規参加団体向けのシステムが必要とする資源を積算し、新たな仮想サーバ環境を追加構築することにより対応可能である。

No.		発生日	状況	発生元	分類	業務	検討事項・問題点	原因・理由	回答・対応・対策	期限	担当者	完了日	参考資料
1	H21.8.24	仕掛	KKC	質問	住民科	住民科	人口は何人ですか？	データベース領域の見積もりをおこなうため	54902（平成22年6月1日現在）				
2	H21.8.24	仕掛	KKC	質問	住民科	住民科	世帯数は何世帯ですか？	データベース領域の見積もりをおこなうため	21814（平成22年6月1日現在）				
3	H21.8.24	仕掛	KKC	質問	住民科	住民科	特徴の人数（非課税者含む）は何人ですか？	データベース領域の見積もりをおこなうため	確認の上、後日回答します。				
4	H21.8.24	仕掛	KKC	質問	住民科	住民科	特徴義務者の数（事業所数）は何人ですか？	データベース領域の見積もりをおこなうため	確認の上、後日回答します。				
5	H21.8.24	仕掛	KKC	質問	住民科	住民科	特徴の人数（非課税者含む）は何人ですか？	データベース領域の見積もりをおこなうため	確認の上、後日回答します。				
6	H21.8.24	仕掛	KKC	質問	住民科	住民科	年間の異動数は何人ですか？	データベース領域の見積もりをおこなうため	確認の上、後日回答します。				
7	H21.8.24	仕掛	KKC	質問	住民科	住民科	管理年度は何年度必要ですか？	データベース領域の見積もりをおこなうため	7年度				
8	H21.8.24	仕掛	KKC	質問	住民科	住民科	雇用所得登録者数は何人ですか？	データベース領域の見積もりをおこなうため	約15000人				
9	H21.8.24	仕掛	KKC	質問	住民科	住民科	住居外の登録者数は何人ですか？	データベース領域の見積もりをおこなうため	約4000人				
10	H21.8.24	仕掛	KKC	質問	住民科	住民科	住居異動の多い月の住居外登録者数は何人ですか？	データベース領域の見積もりをおこなうため	確認の上、後日回答します。				
11	H21.8.24	仕掛	KKC	質問	住民科	住民科	住居異動の多い月の住居外登録者数は何人ですか？	データベース領域の見積もりをおこなうため	確認の上、後日回答します。				
12	H21.8.24	仕掛	KKC	質問	住民科	住民科	普通徴収の期割額と各期の月はいくつですか？		1~4期、6月末、8月末、10月末、翌年1月末 随期あり。翌年2月末、翌年3月末				
13	H21.8.24	仕掛	KKC	質問	住民科	住民科	特徴義務者データの個人コード体系について取り決めのどちらがよろしいですか？		現行システムの取扱いも確認の上、回答します。				
14	H21.8.24	仕掛	KKC	質問	住民科	住民科	特徴義務者データの個人コード付番方法は手動、自動のどちらがよろしいですか？		現行システムの取扱いも確認の上、回答します。				
15	H21.8.24	仕掛	KKC	質問	住民科	住民科	電子入力の委託は行われますか？行われている場合、委託されている課税資料は何か？		確申、市申、給報、公約年金等支払報告書				
16	H21.8.24	仕掛	KKC	質問	住民科	住民科	給報の光ディスク申請は受け付けていますか？		受け付けていません。				
17	H21.8.24	仕掛	KKC	質問	住民科	住民科	証明書に電子印印を使用されますか？	電子印印を使用される場合は、改ざん防止用紙が必要となります。	使用します。				
18	H21.8.24	仕掛	KKC	質問	住民科	住民科	証明書の用紙サイズについてはA4でよろしいですか？		はい。				
19	H21.8.24	仕掛	KKC	質問	住民科	住民科	証明書の複写（控え）は必要ですか？		はい。				
20	H21.8.24	仕掛	KKC	質問	住民科	住民科	証明書の文書について以下のとおり良いですか？変更が必要な場合は、どのような文書にするかご教授下さい。 例)上記のとおり相違無いことを証明します。		上記のとおり、相違無いことを証明します。				
21	H21.8.24	仕掛	KKC	質問	住民科	住民科	証明書番号は必要ですか、また必要な場合は証明書単位での送付ですか？		必要ありません。				
22	H21.8.24	仕掛	KKC	質問	住民科	住民科	証明書番号は支所欄に属する連番としますか？もしくは支所に属せず連番としますか？						
23	H21.8.24	仕掛	KKC	質問	住民科	住民科	証明書に印字する公印は異なる支所で発行してもすべて同じ公印としてよろしいですか？						
24	H21.8.24	仕掛	KKC	質問	住民科	住民科	証明書に印字する証明する者の名前は異なる支所で発行してもすべて同じであると考えてよろしいですか？						
25	H21.8.24	仕掛	KKC	質問	住民科	住民科	証明書の合計所得及び総所得金額については届出後となりませんがよろしいですか？		現行システムも確認の上、後日回答します。				
26	H21.8.24	仕掛	KKC	質問	住民科	住民科	事務納付書の発送（打出し）について当初一括発送ですか？毎期発送ですか？		当初一括発送です。				
27	H21.8.24	仕掛	KKC	質問	住民科	住民科	生活保護法の規定により厚生労働大臣が定める保護の基準における地域の地域区分をご教授下さい。		1級地-2（地域福祉課課認済）				
28	H21.8.24	仕掛	KKC	質問	住民科	住民科	均等割額の金額はいくらですか？		4000円（市3000円、府1000円）				
29	H21.8.24	仕掛	KKC	質問	住民科	住民科	所得割の税率を市町村独自で定めていますか？		定めていません。				
30	H21.8.24	仕掛	KKC	質問	住民科	住民科	前納奨励金がありますか？またその計算式はどのようなものですか？		ありません。				
31	H21.8.24	仕掛	KKC	質問	住民科	住民科	普通徴収の期割金額は1000円単位ですか？		はい。				

図 4-54 ヒアリングシート

カ) データ移行

京都府自治体情報化推進協議会が提供する「ファイル項目移行仕様」に従い、汎用機（ホスト）より移行データが抽出された。なお、データ抽出作業は新規参加自治体からの依頼により、既存事業者（既存システム保守事業者）が実施した。

データ移行の標準仕様である内部コード（Unicode）へのコード変換及び利用者定義文字の移行は既存事業者側の作業としているが、既存事業者側での作業が困難であったため、新規参加自治体の依頼により事業者で作

業を行った。

移行データ抽出作業は導入スケジュールに従い、以下の手順で実施された。

- ① 事業者より既存事業者に対するデータ移行仕様説明（市町村、既存事業者、事業者）
- ② データ移行仕様の確定（市町村、既存事業者）
- ③ データ移行仕様の顧客レビュー（市町村、新規参加団体、既存事業者、事業者）
- ④ データ移行プログラム開発（既存事業者）
- ⑤ テスト用データ移行

データ移行プログラム及び抽出データに問題がないことを検証し、基幹業務支援システムに正常にデータセットアップできることを確認した。

- ◆ テスト用データ抽出（市町村、既存事業者）
- ◆ データセットアップ（事業者）
- ◆ データ検証（市町村）

⑥ システム検証用データ移行

基幹業務支援システムに抽出データをセットアップし、システムが正常に稼動することを確認した。また、新規参加団体によるシステム検証（業務運用の確認）を実施した。

- ◆ システム検証用データ抽出（市町村、既存事業者）
- ◆ データセットアップ（事業者）
- ◆ データ検証（市町村）

⑦ 本稼働用データ移行

- ◆ 本稼働用データ抽出（市町村、既存事業者）
- ◆ データセットアップ（事業者）
- ◆ データ検証（市町村）

⑤～⑥のデータ検証で発生した、抽出データの不具合については、新規参加自治体を通じて既存事業者（データ抽出作業）へ通知し、データ抽出プログラムの変更及び再データ抽出作業が行われた。

キ) システム稼働環境の構築

各業務のヒアリングシートの内容にしたがい、基幹業務支援システムの実装及び運用環境の設定を行った。また、既存システムからの抽出データを基幹業務支援システムに搭載し、移行データの論理チェック（詳細確認はデータ移行作業にて実施）及びシステム動作確認を実施した。

#### ク) 本稼動

平成23年1月より並行稼動（追いかけ入力及び並行入力）を開始した。

#### (4) 実証の考察

「実証の内容」に示した手順にしたがって、導入作業を実施することにより、稼動中の顧客へ影響を与えることなく、平成23年1月より並行稼動を開始した。一連の手順にしたがって作業を行うことにより、新規参加自治体の追加が可能であることが確認できた。

基幹業務支援システム自体は既に3年の稼動実績があり、標準機能範囲における品質は確保されているが、システムが正常に動作するためには、システムが要求する仕様にしたがったデータ移行が必要となる。一旦システムが稼動すると、データ移行のやり直しは不可能な状況となるため、データ移行作業で発生する課題は関係者で共有し、関係者相互の協力のもと安全性を重視した対応を行う必要がある。また、システム導入作業において、データ移行に係る作業に相当の期間と労力を要するため、早い段階での作業着手が必要となる。

仮想化サーバにおける機器構成決定に遅れが発生したが、データ移行作業において代替サーバを利用するなどの対応を行ったため、導入作業全体のスケジュールに影響はなかった。また、本実証実験において、仮想化サーバによる対応が可能であることが実証された。

今回の導入作業において、新規参加自治体との調整により個別改修が発生し、新規参加自治体の費用負担の増加となった。一般的にシステム標準機能を用いて業務運用を行うことが今後の運用費用を含め、システムに関わる費用を軽減することとなる。システム導入における基本方針を徹底するとともに、複数の参加団体で必要とされる機能については、標準機能としての機能追加を検討し、システム標準機能の充実を図るなど、システム面における対応も継続して検討する必要がある。

## 4.4 府・市町村税業務共同化実証

平成20年度に実施された京都府・市町村税務共同化課税業務支援システム基本調査のもと、法人住民税、法人事業税及び地方法人特別税（以下「法人関係税」という。）を対象とし、京都府下において京都府自治体情報化推進協議会が市町村基幹業務の標準パッケージとして推進を行っている「基幹業務支援システム」をベースパッケージとした「共同処理型業務アプリケーション（税務共同化システム）」の開発を平成22年8月より着手した。

本実証では、業務要件及びシステムの機能要件について、各構成団体にて個別に実施されている税業務を業務運用の共同化の観点による見直しを行い、構成団体毎の事務統一に向けた手続（要件定義、仕様決定等）を確認した。

また、構成団体毎でのサービスレベルの格差をなくした均一的な住民サービスの提供と、納税者の利便性向上が期待できることを確認した。

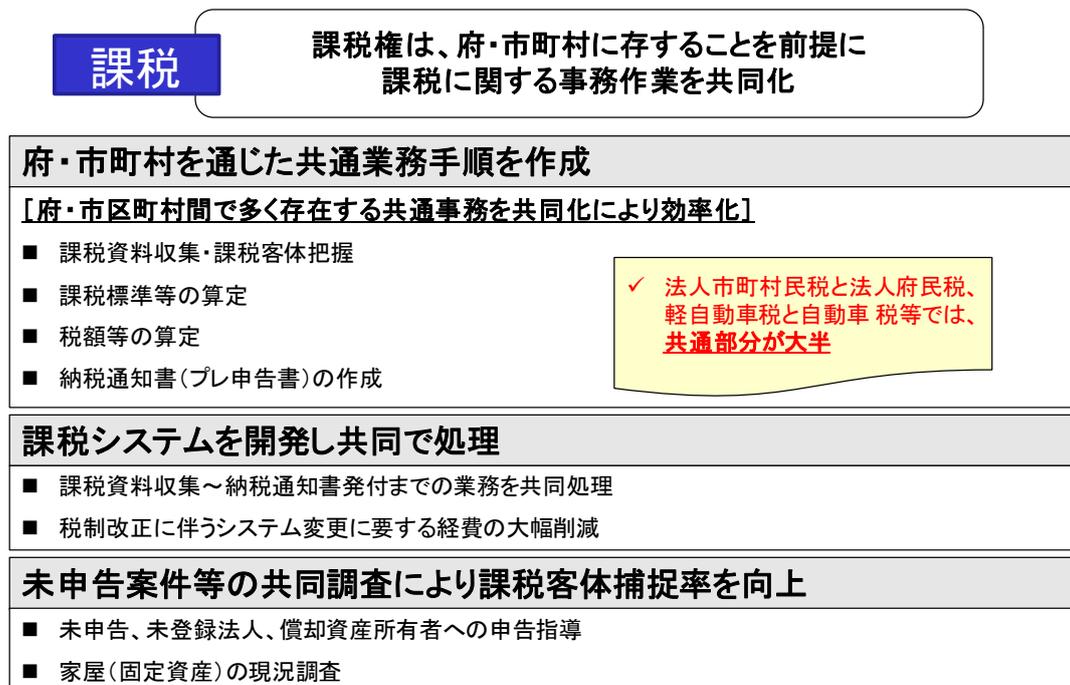


図 4-55 府・市町村税共同化の検討内容

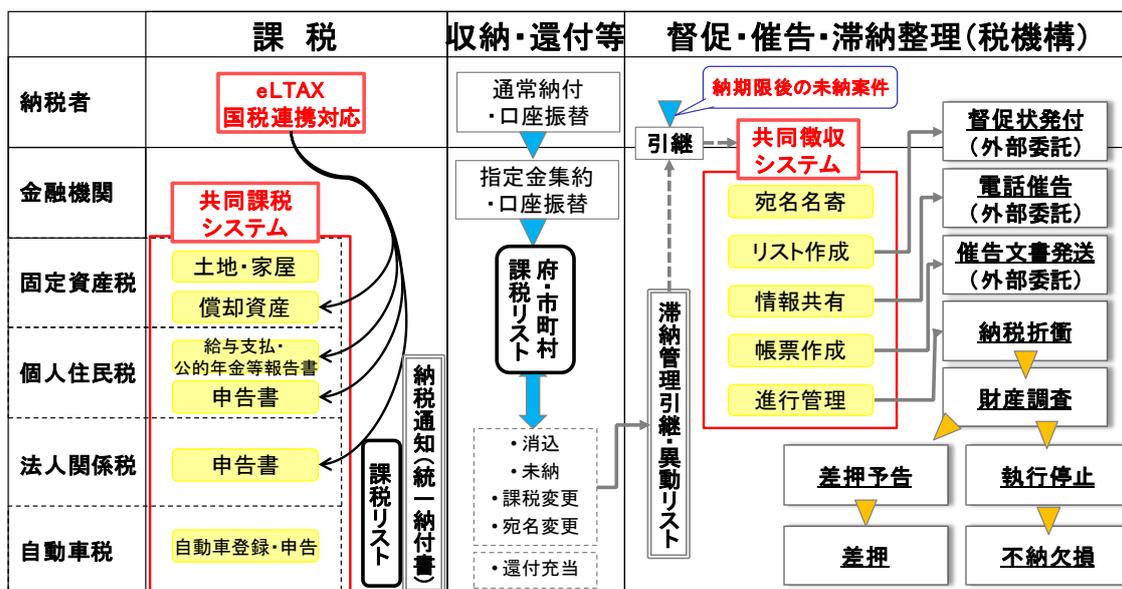


図 4-56 府・市町村税共同化業務・システムのイメージ

### 4.4.1 税業務共同化の経緯

平成18年度に「行財政連携推進会議」にて決定された「税業務の一元化」に向け、平成19年度に学識経験者、実務経験者、行政経験者等外部有識者で構成する「京都府税務共同化推進委員会」を設置し、この委員会で目指すべき税務共同化のあり方及びその具体化に向けて検討され、平成19年12月に「税務共同化に向けた提言」をまとめた。

税業務共同化を広域連合により行う準備のため、平成20年度に京都府・市町村税務共同化組織設立準備委員会が設立され、翌平成21年度に京都地方税機構を設置した。その後、業務支援システム整備検討及び開発に着手し、全構成団体参加による税務共同化システムの構築を実施している。

表 4-104 税業務共同化の経緯

時期	内容
平成18年11月	京都府・市町村行財政連携推進会議において、課税・徴収業務の共同化のあり方について詳細な検討を行うことを確認し、併せて、税業務の一元化が確認された。
平成19年5月	学識経験者、実務経験者、行政経験者等外部有識者で構成する京都府税務共同化推進委員会を設置し、目指すべき税務共同化のあり方及びその具体化に向けての諸課題の検討開始。
平成19年12月	京都府税務共同化推進委員会が、府・市町村を通じて適正な課税と確実な徴収を進め、公平公正で効率的な、府民・納税者に信頼される税務行政の確立を図るため、府税及び市町村税に係る税業務の共同化を推進する「京都府税務共同化推進委員会まとめ（税業務共同化に向けた提言）」を取りまとめ。
平成20年2月～	町村会、市長会において、京都府内の市町村と府の税業務を共同処

時期	内容
	理する広域連合を設立するために必要な検討、調整及び業務支援システム等の整備を行うことを目的とする税務共同化組織設立準備委員会の立上げを確認するとともに、市町村税務担当課長会議において、今後の税務共同化の進め方について確認。
平成 20 年 4 月～	京都市を除く 25 市町村長及び府代表で構成する京都府・市町村税務共同化組織設立準備委員会を設立。準備委員会内に市町村職員・府職員が参加する事務局を設置するとともに、3 つの検討部会を設け、税務共同化に向けて整理すべき項目に対応した検討調整を進める。
平成 20 年 8 月	第 1 回京都府・市町村税務共同化組織設立準備委員会において、広域連合による税務共同化は徴収業務から開始、課税業務はさらに詳細検討することを確認。
平成 20 年 12 月	第 2 回京都府・市町村税務共同化組織設立準備委員会において、広域連合の事務局組織（共同徴収組織）、執行拠点、職員構成、経費等税務共同化案について検討、今後の進め方について申し合わせ。
平成 21 年 4 月	第 3 回京都府・市町村税務共同化組織設立準備委員会において、広域連合規約案、事業計画案、今後の手順について協議、同意。
平成 21 年 8 月	設立許可。 広域連合長選挙の執行。 京都地方税機構の事務所を開設。

## 4.4.2 税業務共同化の考え

府・市町村の税業務共同化に向け、京都府税務共同化推進委員会が平成 19 年 12 月にとりまとめた「京都府税務共同化推進委員会まとめ（税業務共同化に向けた提言）」を紹介する。

### （1） 税業務共同化の趣旨と目的

府税及び市町村税に係る税業務の共同化は、府・市町村を通じて適正な課税と確実な徴収を進め、公平公正で効率的な、府民・納税者に信頼される税務行政を確立することを目的とする。

この目的の遂行にあたっては、特に次のことに留意をする。

税業務の共同化により、業務の標準化等を進め、公平な課税と、効果的な徴収業務を確立して貴重な自主財源の徴収率向上を実現するとともに、府民の視点から、簡素でわかりやすい税の組織、業務体系を構築する。

具体的には、以下の点に特に留意をしながら共同化を推進する。

- ① 複数税目の申告・納付等窓口の一本化により納税者の利便性向上を図る。
- ② 納付相談・税務相談、不服申立て等の処理手続を整備し、府民の声に迅速に対応するなど、納税者への対応（納税者対応）の向上を図る。
- ③ 重複する税業務を整理するとともに、課税・徴収業務の標準化を進め、公平な課税と効果的な徴収を実現する。

④ 税業務の遂行について徹底したコストの圧縮を図る。

税業務の共同化により、地方分権の推進に向けて自主財源である税収を確保し、更に国から地方への税源移譲に 대응できる税務執行体制を構築する。

(2) 税業務共同化の具体的な内容

ア) 府・市町村を通じた税業務の共同化（新しい税業務体系の構築）

府税・市町村税に係る税業務の共同化にあたっては、次の3つの業務形態に整理できる。

A. 府内1箇所での一本処理業務

大量反復作業や専門性が高い業務等、一本処理が最も効果的・効率的で、利便性を高める業務（申告の窓口一本化、賦課通知の文書発送、電話催告、特別機動整理案件、不服申立て・訴訟、システム管理等）などは、一本処理を進める。

B. 広域的な共同処理業務

共同処理が合理的、かつ効果的・効率的な業務で、適宜、現地現場での作業が必要な業務（課税客体の把握、納税折衝、滞納処分、家屋評価業務等）などは、地域における広域的な共同処理を進める。

C. 市町村庁舎等で処理する業務

住民との対面でのやりとりが必要な業務や知事・市町村長が名義人（行政庁）として行うべき業務（納税証明書の発行、固定資産課税台帳の閲覧、課税権に基づく賦課決定等）については、各市町村庁舎等で対応することとする。

イ) 税業務についての原則的な標準化、一本化（手続、帳票の様式、処分基準等）

納税者への公平・公正、納税者の利便、行政の効率化といった視点から、市町村間で税業務の取扱い等の基準に相違が生ずることのないように業務や書式等の標準化を図る。

ウ) 徹底した業務見直しによる効率化の推進

文書催告の共同作成・共同発送、電話催告の共同センター化、消込業務の効率化、補完的・大量反復的作業の外部委託化等により、徹底した業務の見直しを行い、税務行政の合理化、効率化を図る。

### (3) 共同で処理する業務の範囲等

#### ア) 共同化の範囲

税業務の共同化は、地方自治法、地方税法等の現行法の範囲（枠）内で進められるため、次のような一定の制約があることに留意をすべきである。

- ① 地方団体の課税権は立法と執行に区分されるが、立法権（税条例の制定）は地方税法により地方団体に与えられたものであり、税業務の共同化の範囲には含まれない。
- ② 執行権は、課税と徴収に区分されるが、申告納税方式における更正・決定、賦課課税方式における賦課決定等の租税債権を確定する権限は、地方税法において地方団体の長に存することから、賦課決定等（行政処分）自体は共同化の対象には含まれない。
- ③ 還付、充当については、課税権の行使に密接に関係しており、地方税法上、地方団体の長に権限が存することとされていることから、共同化の対象には含まれないものと解される。
- ④ 犯則取締りに係る業務については、一定の手続きを除き、告発又は通告処分等は共同化の対象に含まれないものと解される。

上記以外の課税資料の収集、課税標準の算定等に係る課税業務や催告、折衝、差押え等の徴収業務は、原則として共同処理が可能と解される。

#### イ) 税業務共同化の基本業務

税業務の共同化における課税から収納、滞納整理に至る基本的な業務は、次のとおりである。

##### A. 課税データの作成

法人に関係する主な税目は、府税・市町村税を問わず申告等を一括受付する。更に、電子申告の活用を促進して、ワンストップサービスを実現する。

- ① 固定資産税に係る償却資産の申告を一括で受付・入力し、納税通知書を共同作成・発送する。
- ② 給与支払報告書を一括で受付・入力し、市町村にデジタルデータを配信する。
- ③ 法人市町村民税、法人二税（法人府民税・法人事業税）の申告を一括で受付・入力し、プレプリントを送付する。

その他の税目についても、データ連携や入力等の一括委託によりコスト削減を実現する。

## B. 収納データの作成

納付された税金の領収済データ作成については、省力化・迅速化を実現する。

- ① 領収済通知書のフローを見直し、消込データ作成を一括委託する。
- ② 消込データは各自治体システムに登録し、窓口業務（収納、還付、証明等）に対応可能な体制を確保する。

## C. 滞納データの管理

納期限後の未納案件は、一元的に共同処理を実施する。

- ① 支援システムを活用し、電話催告・文書催告等の外部化により大量案件を圧縮する。
- ② 職員による財産調査、差押えを実施し、徴収率の向上を実現する。

## ウ) 共同化の対象となる税目

徴収業務（収納、滞納整理）は、府税・市町村税を問わず業務に共通性があり、また、対象者に重複があることから、全税目を対象として共同処理することが望ましい。

課税業務は、府税・市町村税の各税目で業務フローが異なっており、各税目別に共同処理部分（市町村と府の共同処理、市町村相互の共同処理）を具体化することが必要である。以下のように共同処理を進める。

- ① 法人関係税については、申告、届出の窓口一本化から、是認、更正・決定処理、調査までを共同化する。
- ② 個人住民税については、給与支払報告書等の提出窓口一本化から、納税通知書発送までを共同化する。
- ③ 固定資産税（償却資産）については、申告の窓口一本化から、納税通知書発送までを共同化する。
- ④ 固定資産税（土地・家屋）、不動産取得税については、家屋評価の共同化、異動データ等の収集を共同化する。
- ⑤ 都市計画税については、固定資産税課税データの活用から、納税通知書発送までを共同化する。
- ⑥ たばこ税については、申告の窓口を一本化する。
- ⑦ その他の税目は、滞納整理業務からの共同処理を図る。

なお、府税のうち個人事業税、自動車税、軽油引取税等については現状でも広域的な処理を実施しているが、一層の合理化等を進めることが必要である。また、市町村税のうち軽自動車税、鉱産税、入湯税等についても、一層の合理化を目指して、府内一本処理や隣接市町村の共同処理等の工夫が必要である。

国民健康保険税（料）の徴収業務については、共同組織のもとでの徴収を希望する市町村分を実施する方向で共同化を進める。

## エ) 不服審査等の共同化

年間の不服審査申立件数（25市町村・府合計）は、近年60件～70件で推移しているが、その7～8割を固定資産価格に関する固定資産評価審査委員会への不服申立てが占めている。

不服審査については、共同組織の枠内で、公正な審査と、府民の声に迅速かつ十分に説明責任を果たせる、専門的で効率的な審査体制を整備する。税務相談、訴訟等についても共同組織の枠内でのサポート体制を構築する。

不服審査については、3つの処分類型（課税決定等、固定資産税に係る課税台帳価格、差押え等の滞納処分）に応じて、以下のように、共同化の工夫が必要である。

- ① 課税決定等に係る市町村長・知事あての不服申立てについては、共同組織の専門部署で集中的に支援することが審査の中立性、専門性の確保に資することとなる。
- ② 課税台帳価格に係る固定資産評価審査委員会への不服申立てについては、同委員会の共同設置が可能であることから、統一的な委員会を設置する。
- ③ 差押え等の滞納処分に係る不服申立てについては、共同組織（広域連合）の長が名宛人となるものであり共同組織の専門部署で集中的に処理する。
- ④ 共同組織の専門部署で集中的に処理・支援することにより、名宛人の違いにかかわらず実質的な不服審査の共同化を図る。

不服申立手続について、納税者の利便性に配慮すること（申立窓口、審理場所等）や、申立てに至るまでの苦情対応段階において十分に説明責任を果たすことが重要である。

## オ) 広報、研修等の共同化

課税・徴収業務を共同化するとともに、税務広報や専門的・体系的な税務研修の共同化、支援システム業務の共同化により、税務行政における専門性の向上、効率化を図る。

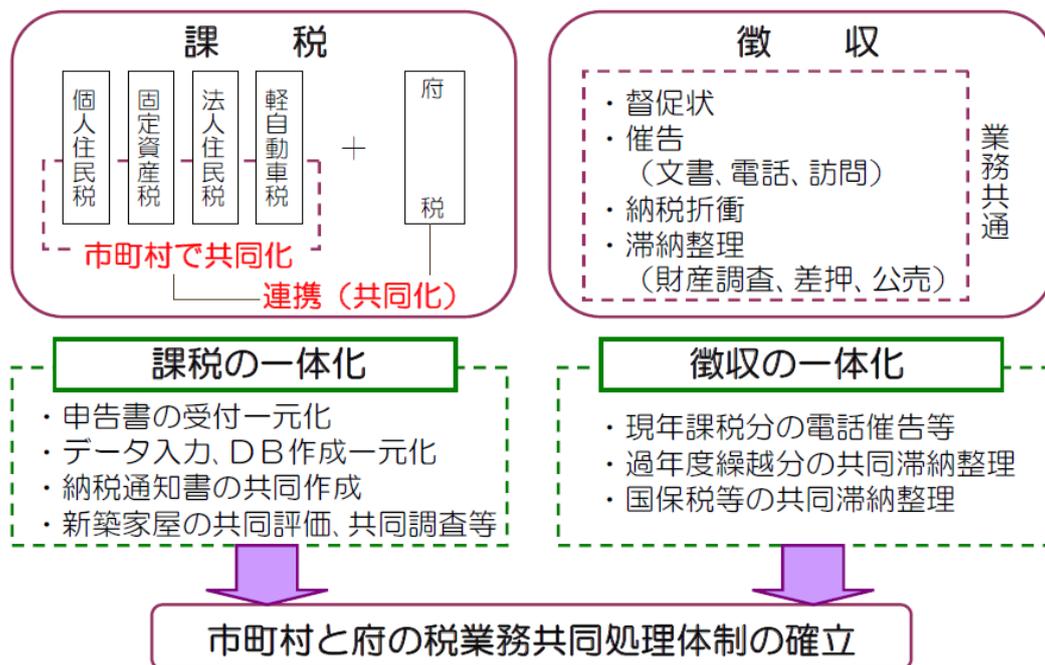


図 4-57 税業務の共同化範囲イメージ

### 4.4.3 法人関係税等支援システムによる共同化実証

#### (1) 概要・目的

平成20年度に実施された京都府・市町村税務共同化課税業務支援システム基本調査のもと、法人関係税を対象とし、京都府下において京都府自治体情報化推進協議会が市町村基幹業務の標準パッケージとして推進を行っている基幹業務支援システムをベースパッケージとした「共同処理型業務アプリケーション（税務共同化システム）」の開発を平成22年8月より着手した。

実証内容としては、業務要件及びシステムの機能要件について、各構成団体にて個別に実施されている税業務を業務運用の共同化の観点による見直しを行うことで、構成団体毎の事務統一に向けた手続き（要件定義、仕様決定等）を確認する。

また、構成団体毎でのサービスレベルの格差をなくした均一的な住民サービスの提供と、納税者の利便性向上が期待できることを確認した。

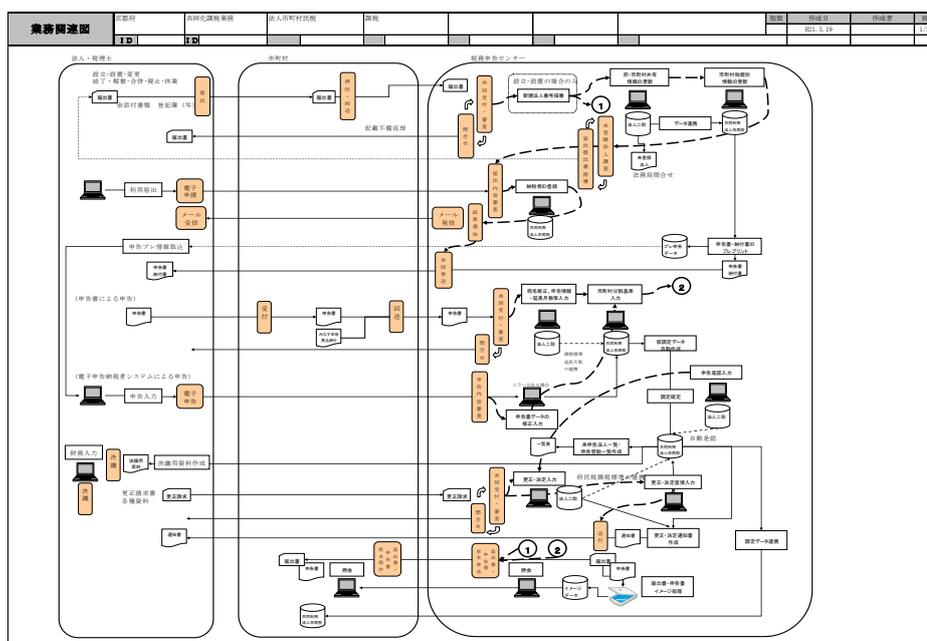


図 4-58 共同課税基本調査 法人関係税 業務関連図（平成 21 年度 調査検討）

#### (2) 業務要件検討手法

現行の市町村での業務運用フローをもとに、共同化による業務運用（業務フロー）の見直しと共同化により必要となる業務要件の検討を行った。その結果をもって、ベースパッケージである基幹業務支援システムでの保有機能との比較を行うことで、機能拡張などのシステム改修にかかる改修費用の軽減が可能と考えた。

また基幹業務支援システムの法人住民税機能を、構成団体毎に独立した「共同利用型法人住民税システム」として導入し、税務共同化システムで構築する機能要件との分担を検討することで、基幹業務支援システムで保有する機能を

最大限に活用可能となると考えた。

さらに府法人二税に関する運用については、市町村民税としては存在しない法人事業税も必要となることから、既存の京都府税務トータルシステムを利用した運用を前提とすることで、開発機能の削減を目指した。

これらの観点のもと要件定義から仕様確定に到る一連の作業を通して、税務共同化システムとして必要となる業務要件の検討と、導入における税業務運用の標準化手法を確認した。

ア) 要件定義

A. 共同化機能要件の抽出

基幹業務支援システム導入ユーザでの法人市町村民税業務の運用フローをベースとし、以下の観点で共同化業務フローの作成を実施した。

- ・ 構成団体で実施すべき業務か、税機構で共同化すべき業務か
- ・ 税機構において共同化する場合、現行の構成団体での運用にどのような影響が発生するか
- ・ 共同化することで、業務運用にかかるタイムラグやデータの不整合はおきないか

その中で府法人二税での申告受付・申告書データ化等の業務運用も共同化することで、都道府県と市町村にわたる業務運用の共同化を実現し、業務運用の効率化と情報の一元化を考慮した。

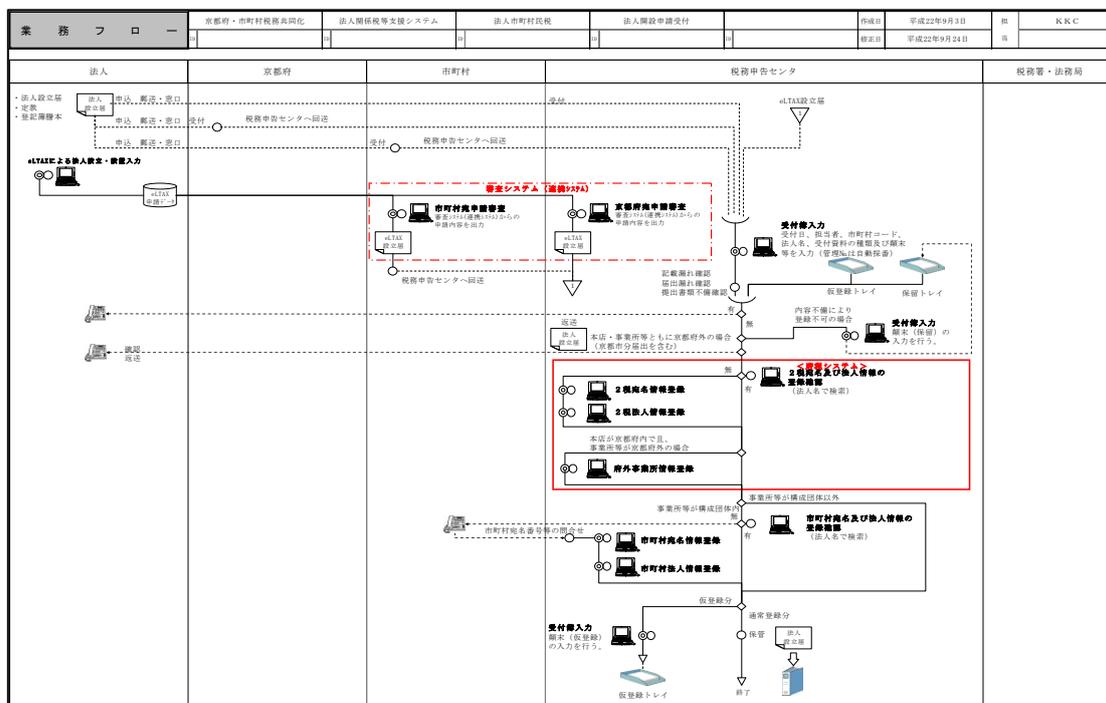


図 4-59 法人関係税等支援システム 業務フロー（抜粋：法人開設申請受付）

## B. 機能要件の確定

- a. 作成した業務フローについて、京都府との協議を行い、以下の業務要件の追加を行った。

### 【追加となった業務要件】

- ・ 府法人 2 税として入力された宛名情報や法人基本情報を法人市町村民税へ連携することによる情報の一元化、入力作業の効率化
- ・ 京都府と市町村で相違のあった「みなし申告業務」の運用統一化

- b. 共同化による業務フローについて、以下の手続により各構成団体へ説明。

平成22年11月2日京都府並びに代表市町村（注1）にて、事前調整

平成22年11月4日法人関係税業務構築グループ会議（注2）開催

（注1）宇治市・八幡市・大山崎町の三市町

（注2）京都府及び、京都市を除く府下25市町村の税務担当所管課において法人関係税業務の担当職員で構成される税務共同化に関する具体的検討を行う組織

## C. 業務分担を説明

平成22年12月3日に再度開催された法人関係税業務構築グループ会議において、業務フローもとに、税機構と構成団体での業務分担を整理した「法人関係税課税事務分担表」を配布・説明を行った。

## D. 課題検討

今回検討した業務要件の検討において、課題となる事項については、課題一覧を作成した。それらの課題に関する整理を行い、解消に至っていない課題に関しては、システム開発においての影響有無の観点から、以下の分類で優先度をつけた。

- PA : システム上影響有（優先度 高）
- PB : システム上影響有（優先度 中）
- PC : システム上影響有（優先度 低）
- PD : システム上影響無・運用にのみ影響

表 4-105 課題事項一覧表（抜粋）

課題事項一覧表		作成日	平成22年9月8日	作成者	KKC							
		修正日	平成22年12月14日	修正者	KKC							
No.	記入日	記入者	業務	システム/運用	内容	解決方法			課題解決	優先度	確認日	関係団体
						内容	済	未済				
1	平成22年9月10日	KKC	法人税務申告受付	システム	法人税務申告受付システム 法人税務申告受付システムを介して法人税務申告受付を行う。	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	□	■	PA	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	関係団体 作成団体	
2	平成22年9月10日	KKC	法人税務申告受付	システム	法人税務申告受付システム 法人税務申告受付システムを介して法人税務申告受付を行う。	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	□	■	PA	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	関係団体 作成団体	
3	平成22年9月10日	KKC	法人税務申告受付	運用	法人税務申告受付システム 法人税務申告受付システムを介して法人税務申告受付を行う。	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	□	■	PD	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	関係団体 作成団体	
4	平成22年9月10日	KKC	法人税務申告受付	運用	法人税務申告受付システム 法人税務申告受付システムを介して法人税務申告受付を行う。	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	□	■	PD	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	関係団体 作成団体	
5	平成22年9月10日	KKC	法人税務申告受付	システム	法人税務申告受付システム 法人税務申告受付システムを介して法人税務申告受付を行う。	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	□	■	PD	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	関係団体 作成団体	
6	平成22年9月10日	KKC	法人税務申告受付	運用	法人税務申告受付システム 法人税務申告受付システムを介して法人税務申告受付を行う。	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	□	■	PD	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	関係団体 作成団体	
7	平成22年9月10日	KKC	法人税務申告受付	システム	法人税務申告受付システム 法人税務申告受付システムを介して法人税務申告受付を行う。	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	□	■	PD	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	関係団体 作成団体	
8	平成22年9月10日	KKC	法人税務申告受付	運用	法人税務申告受付システム 法人税務申告受付システムを介して法人税務申告受付を行う。	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	□	■	PD	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	関係団体 作成団体	
9	平成22年9月10日	KKC	法人税務申告受付	システム	法人税務申告受付システム 法人税務申告受付システムを介して法人税務申告受付を行う。	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	□	■	PD	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	関係団体 作成団体	
10	平成22年9月10日	KKC	法人税務申告受付	システム	法人税務申告受付システム 法人税務申告受付システムを介して法人税務申告受付を行う。	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	□	■	PD	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	関係団体 作成団体	
11	平成22年9月10日	KKC	法人税務申告受付	システム	法人税務申告受付システム 法人税務申告受付システムを介して法人税務申告受付を行う。	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	□	■	PD	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	関係団体 作成団体	
12	平成22年9月14日	KKC	法人税務申告受付	運用	法人税務申告受付システム 法人税務申告受付システムを介して法人税務申告受付を行う。	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	□	■	PD	法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。法人税務申告受付システムを介して法人税務申告受付を行う。	関係団体 作成団体	
13	平成22年9月13日	KKC	申告書送付	システム/運用	申告書送付システム 申告書送付システムを介して申告書送付を行う。	申告書送付システムを介して申告書送付を行う。申告書送付システムを介して申告書送付を行う。申告書送付システムを介して申告書送付を行う。	□	■	PD	申告書送付システムを介して申告書送付を行う。申告書送付システムを介して申告書送付を行う。申告書送付システムを介して申告書送付を行う。	関係団体 作成団体	

(3) 実証の考察

前述までの手順にて、共同化業務における業務要件検討作業を実施した。

共同化による業務要件に関しても、基本的には現行の各構成団体での運用に準じるものであり今回の業務要件の検討においては、ベースパッケージである基幹業務支援システムを導入するユーザにおける業務フローが存在したことで、要件検討において大幅に期間の短縮が可能となった事や、システムに関する開発機能の検討をスムーズに実施することが出来た。

この事は、今後京都府以外の自治体において課税業務の共同化を検討する上で、今回作成した業務フローをベースとした要件分析を行うことで、同様の効果を得られることを確認できた。

また、電子申告により申告された情報を活用するため、共同利用型審査システムからデータ連携を行うことを前提としてシステム構築することで、電子申告の普及により、大幅な業務の効率化が期待できることを確認した。

また今回要件検討においては、税務共同化システム単独で共同化に関する全ての業務要件を対象としたシステム化ではなく、税機構で共同化して行うべき業務と、構成団体毎で行うべき業務を明確に分割することで、開発範囲の削減を目指した。

構成団体にて行うべき業務に関しては、基幹業務支援システムの法人市町村民税機能を各構成団体分準備し利用すること、府法人2税に関しては、既存の京都府税務トータルシステムをそのまま運用することで、既存のシステムを最大限に有効利用することが可能となり、あわせて開発費用の大幅な削減が可

能となった。

その反面、複数のシステムの連携を行うことが必要となり、システム間での連携内容を十分に検討する必要があった。

加えて業務運用面においても、複数のシステムを利用することが必要となり、今後、本番稼動に向けた運用テスト等の状況に応じて、運用担当者の業務分担や体制検討を行うなど、システムの運用形態も含めた検討が不可欠であることが確認できた。