



INFORMATION ANYWHERE

iAnywhere
A SYBASE COMPANY

Technical Guide

リモートサポートを考慮した データベースアプリケーション構築

ポール・ホラン著



目次

はじめに	3
VCIについて.....	3
アクション・アイテム.....	4
サイトの準備	5
リモートコントロールソフトウェアによるインターネット接続の確立.....	5
定型業務の自動化	7
無人設計データベースのバックアップ	7
重要なサーバ統計の自動監視	9
データベースの診断.....	12
“Remember_Last_Statement”エンジンプロパティの設定	12
“Request_Level_Logging”の実行.....	12
ブロックされた接続の検出.....	13
ロングランクエリの検出	13
データベースのローカル・コピーの取得	15
まとめ.....	16
本書の著者について.....	17
法的注意	18

はじめに

「優れたネズミ取りを作れば、人々はあなたの家の前に列をなすであろう」という金言がありますが、ソフトウェアビジネスでは、リモートの顧客用の特殊な処理や機能をサポートする組み込みアプリケーションがこの「ネズミ取り」になることがよくあります。組み込みアプリケーションとは、その全アーキテクチャ（ハードウェア、アプリケーションソフトウェア、DBMS）が顧客のサイトに実装されるアプリケーションのことです。組み込みアプリケーションの設計と開発についてはよく知られていますが、配布後の組み込みアプリケーションへのサポートの提供という課題は見過ごされることが多いのが実情です。

本書ではこの課題について、Video Communications, Inc. (VCI) による実際のアプリケーション開発の経験に基づき、リモートサポートを考慮したデータベースアプリケーション構築の要点を解説します。

アプリケーションの実装前に注意深く計画を立て、準備を怠らなければ、リモートによる組み込みアプリケーションのサポートに伴う困難を大幅に減らすことができます。ネズミ取りを求めてあなたの家を訪れる人々を失望させないために、本書をお役立てください。

VCIについて

マサチューセッツ州スプリングフィールドに本拠を置くVCIは、テレビ局とケーブルネットワーク向けソフトウェアの分野をリードする独立系ソフト開発企業です。1995年、VCIは主力ソフトウェア製品のSTARS IIを完全に再設計するプロジェクトを開始しました。その目的は既存STARS IIのDOSインタフェースをWindows互換のGUIに変更し、MUMPSデータベースをiAnywhereのSQL AnywhereリレーショナルDBMSに切り替えることでした。STARS IIとその後継機種であるSTARS II+は完全な組み込み型アプリケーションであり、VCIのソフトウェアのコンポーネントがVCIやサードパーティのサービス・プロバイダにホストされることはありません。

STARS II+は現在米国とカナダの100以上のテレビ局とケーブルネットワークに実装されていますが、そのほとんどにはフルタイムのITスタッフが常駐していません。

アクション・アイテム

組み込みアプリケーションを購入する顧客は、問題が見つければ、その原因が開発企業とDBMSプラットフォームなどのサードパーティのどちらにあるかに関わらず、開発企業側がサポートの求めに応じるのが当然と考えるものです。そのため開発者にとって、優れた機能を持つ一方でオンサイト・サポートをほとんど必要としないデータベース・プラットフォームの選択は重要です。またフルタイムの管理者がいなくともある程度のサポート業務は実行しなくてはなりません。通常は開発企業のサポートおよびエンジニアリングスタッフには、こうしたアプリケーションへリモートで継続的かつリアルタイムにアクセスすることはできないですし、すべてのサイトをチェックするだけの時間ありません。

上記の課題をクリアするために重要なアクション・アイテムは、大きく分けるとクライアントのサイトへのアプリケーションのリリース前の段階とシステム運用の段階でそれぞれ二つづつ、合わせて四つあると考えてよいでしょう。

<リリース前の段階>

- サイトの準備
クライアントのサイトへアプリケーションをリリースするずっと前から顧客サポート提供の準備を行っておく必要があります。クライアントのサイトにあるリモートマシンへアクセスする許可を取得し、またその手順を開発しておきます。さらにリモートアクセス用のソフトウェアパッケージを選定し標準化しておく必要もあります。
- 定型業務の自動化
バックアップやディスクとデータベースのデフラグといった定期管理タスク、またディスクドライブ容量の監視といった主体的なタスクを自動化するソフトウェアルーチンを構築しておく必要があります。

<運用開始後の段階>

- データベースの診断
第一線のサポート要員が収集した問題の切り分けと診断を行います。
システムのスローダウンや停止として表面化する問題は、実際には異なる複数の原因による 경우가多々あります。こうした原因の調査に費やす時間を削減することが、サポートを成功に導く最も重要なポイントです。
組み込みSQL Anywhereデータベースに関する問題の診断に役立つコマンドは複数用意されています。この中には、製品のパフォーマンスにほとんど影響を及ぼすことなく、顧客サイトのデータベースに対して直接実行できるものもあります。
- データベースのローカル・コピーの取得
これは顧客サイトのデータベースで直接解決できない問題の「エスカレーション」に相当します。問題の性質上、詳細な分析のためにデータベースのコピーをラボに持ち帰る必要が生じることもあるはずですが。

以下の章では上記の四点にしぼってアプリケーション開発の要点を解説します。

サイトの準備

リモートコントロールソフトウェアによるインターネット接続の確立

リモートデータベースサーバへの接続とアクセスは最も重要であると同時に最もデリケートな問題でもあります。組み込みデータベースソリューションの導入にあたっては、サポートおよびエンジニアリングスタッフがリモートでデータベースサーバへアクセスし管理することを顧客に認めてもらうことが重要です。サーバのコマンドラインパラメータの設定、データベースファイルのバックアップコピーの圧縮と転送、動作が遅いクエリの診断といったベーシックなサポート業務は電話で行なえるものではありません。ワイドバンドベースの高速のリモートコントロールソリューションがどうしても必要になるのです。

エピソード

90年代の中盤には、これはスタティックなIPアドレスを与えたサーバをファイアウォールの外あるいはDMZに設置することを意味しました。当然ながらこれは顧客のデータを無防備でインターネットにさらけ出すことになるので危険な上、効率も劣ります。VCIの最初のリモートコントロールソフトウェアの実装では、56Kモデムダイヤルアップ回線上でSymantec pcAnywhereを使用しました。これはセキュアではあったもののどうしても遅く、まったく非効率的なオプションでした。今日ではきわめて安全で低コスト(あるいはフリー)のリモートデスクトップコントロールソリューションが複数あり、サーバをNATレイヤ、ルータ、ファイアウォールの背後に設置できるようになりました。VCIは現在Webex社のSmartTechソフトウェアを標準実装し、各クライアントサイトに実装されたデータベースサーバへの即時かつ高速のアクセスを実現しています。

リモートコントロールパッケージについて検討する必要があるその他の重要な機能を以下に示します。

- NAT、ルータ、ファイアウォールの背後のマシンへのアクセス
- ファイル転送
- リブートと再接続
- 画面のスケーリング
- チャット機能
- SSL暗号化
- アクセス監査証跡

以下に現在市場で入手可能な各種のリモートコントロールソフトウェアを簡単にまとめました。

表1. リモートコントロールソフトウェア

ベンダ	製品名	説明	ホームページ
Webex	SmartTech	各企業固有のニーズに合わせて異なる製品を提案	http://www.webex.co.jp/jp/index.html
NetworkStreaming	SupportDesk	Hosted、Professional、Liteの三つのレベルが購入可能	http://www.networkstreaming.com
Citrix	GotoAssist	オプションでRemedyベースのヘルプデスク追跡システムを統合	http://www.gotoassist.com
3amLabs	LogMeIn IT Reach	リモートマシンを監視できるリアルタイムの「ダッシュボード」を提供	https://secure.logmein.com/
RealVNC	VNC Enterprise Edition	Enterprise版は低価格、クロスプラットフォームを（WindowsとUnix）サポート	http://www.realvnc.com

定型業務の自動化

「組み込みソフトウェアパッケージを顧客に納品してしまえば、その後は顧客が管理及び保全タスクを理解し、それを厳格に実行することを定め、そして実際に正しく実行するだろう」などと考えるはいけません。こうしたタスクをできる限り明確化し、自動化し、簡素化する責任はベンダにかかっているのです。幸いにもSQL Anywhereバージョン9.0.2は管理の手間がほとんどかからず、また主体的な保守タスクの開発を支援するための機能を数多く提供します。

SQL Anywhereは、.BATファイルやUnixシェルスクリプトにスクリプト化してスケジューリングできるDBBACKUPやDBVALIDなどのコマンドラインユーティリティをフルセットで提供します。.BATファイルには、接続パラメータ（UIDとPWD）がオープンテキストに入力できるという弱点がありますが、この潜在的なセキュリティ・ホールは以下に示す複数の方法で回避できます。

- dbfhideユーティリティを利用してコマンドラインオプションを格納する環境設定ファイルの内容を簡単に暗号化する
暗号化後の環境設定ファイルは“@”パラメータeg. dbbackup @config_fileとして使用可能です。
- バックアップ操作用に別途データベース・アカウントを作成する
このアカウントへのアクセス許可をGRANT REMOTE DBAにのみ与えれば、悪意のあるユーザーがアクセスに成功しても、できることはほとんどありません。
- DBTOOLS APIライブラリを利用してWindowsプラットフォーム上のこれらのユーティリティをプログラム制御する
先進的な手段ですが、これらのAPIを起動するC/C++ユーティリティを書くことで、その実行を内部的に制御する手段を提供できるはずです。

おそらくSQL Anywhereで最も強力かつ柔軟な機能はイベントスケジューラです。SQL Anywhereは、イベントをスケジュールされた間隔で定期的に起動したり、SQL Anywhereが検出した特定の条件とリンクしたり、リクエストに応じて実行するようプログラムできる内部スケジューラを備えています。これをデータベース中の特定のテーブルと関連付けられていないトリガと考えればよいでしょう。イベントの利用には、.BATファイルあるいはカスタムメイドのアプリケーション・ラッパーと比べて明らかに有利な点がいくつかあります。

- イベントハンドラスクリプトがC/C++や.BATファイルシンタックスではなくSQLで書かれる
- イベントハンドラはファイルシステムではなくデータベースに保存される
- 外部の認証を必要としないので.BATファイルにパスワードが書き込まれない

イベントとイベントスケジューラは、アプリケーションの自己診断機能のベースとしても利用できます。

無人設計データベースのバックアップ

リカバリー手順の設計と実装は、保守対象のアプリケーションの性質に応じて大きく異なります。考えられるすべての要件の組み合わせに対応できるスキーマは存在しません。多大なトランザクションを扱うアプリケーションではログのバックアップを頻繁にとる必要があるでしょうが、別種のアプリケーションでは夜間の完全バックアップで十分かも知れません。

NOTE

強靭なバックアップ&リカバリープロシージャの開発については「SQL Anywhere Studio 9 デベロッパーズガイド」(Breck Carter著、翔泳社)の第9章に優れた解説があります。

以下に示すのは最もシンプルなバックアップの例で、夜間イベントとしてサーバのファイルシステム中の任意のフォルダーの完全なイメージバックアップを実行し、アクティブトランザクションのログを廃棄するものです。

```
CREATE EVENT "nightlyFullBackup" ENABLE AT ALL
HANDLER
begin
  BACKUP DATABASE
  DIRECTORY 'd:¥¥database¥¥backup'
  TRANSACTION LOG TRUNCATE
end
go
ALTER EVENT "nightlyFullBackup"
  ADD SCHEDULE "nightlyFullBackup_1"
  START TIME '00:00:01' ON
  ('Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday')
go
```

以下に示すのは、settingsテーブルのデータベース列からバックアップのロケーションを取得し、その値をBACKUP DATABASE文に解釈してEXECUTE IMMEDIATEを実行することで、上述のバックアップを動的なものにした例です。

```
CREATE EVENT "nightlyFullBackup" ENABLE AT ALL
HANDLER
begin
  declare backupFolder varchar(255);
  declare backupCmd varchar(255);

  set backupFolder =
    (select folderName
     from dbOptions );

  set backupCmd =
    `BACKUP DATABASE ` ||
    ` DIRECTORY ` || backupFolder ||
    ` TRANSACTION LOG TRUNCATE`;
  EXECUTE IMMEDIATE WITH ESCAPES OFF backupCmd ;
end
go
```

高トランザクション環境では、平日に定期的な間隔でログだけのバックアップを実行する必要があるはず
です。以下は、特定の間隔でイベントを繰り返し実行するようにイベントスケジューラを設定した例です。

```
CREATE EVENT "hourlyLogBackup" ENABLE AT ALL
HANDLER
begin
  declare backupFolder varchar(255);
  declare backupCmd varchar(255);
  set backupFolder
    =(select folderName from
      dbOptions);
  set backupCmd =
```



```
'BACKUP DATABASE ' ||  
  ' DIRECTORY ' || backupFolder ||  
  ' TRANSACTION LOG ONLY ' ||  
  ' TRANSACTION LOG RENAME MATCH ' ;  
EXECUTE IMMEDIATE WITH ESCAPES OFF backupCmd ;  
  
end  
go  
ALTER EVENT "hourlyLogBackup"  
  ADD SCHEDULE "hourlyLogBackup_1"  
    BETWEEN '10:00:00' AND '18:00:00'  
    EVERY 1 HOURS ON  
  ('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday')  
Go
```

重要なサーバ統計の自動監視

ディスクスペースの減少やデータベースファイルの過度の断片化などの重要なサーバリソースを監視するイベントハンドラを書くことができます。監視対象アイテムが断片化したテーブルやインデックスのように修復可能なもの場合は、修復イベント自体を記述できます。

以下に示すのは日曜日の朝4:00amに起動し、第四レベルを越えて増加したインデックスを検出するイベントです。検出されたインデックスはただちに再編されます。またこのイベントはストアドプロシージャ中でSELECT文を記述できるというSQL Anywhereのもう一つの強力な機能を示す例にもなっています。

```
CREATE EVENT "indexReorg" ENABLE AT ALL  
HANDLER  
BEGIN  
  declare tName varchar(60);  
  declare iName varchar(60);  
  declare iType varchar(60);  
  declare stmt varchar(60);  
  declare err_notFound exception for  
    sqlstate value '02000';  
  
  declare indexLevels no scroll cursor for  
    SELECT  
      tableName,  
      indexName,  
      indexType  
    FROM sa_index_levels()  
  WHERE  
    levels >= 4;  
  
  OPEN indexLevels;  
  indexLoop:  
  LOOP  
    FETCH NEXT indexLevels  
    INTO tName,  
        iName,  
        iType;  
    IF sqlstate = err_notFound then  
      LEAVE indexLoop  
    END IF;  
  
    SET stmt='REORGANIZE TABLE ' || tName;  
    CASE upper(iType)
```

```
WHEN 'PKEY' THEN
    SET stmt=stmt || ' PRIMARY KEY;';
WHEN 'FKKEY' THEN
    SET stmt=stmt || ' FOREIGN KEY ' || iName || ';';
WHEN 'NUI' THEN
    SET stmt=stmt || ' INDEX ' || iName || ';';
END CASE ;

MESSAGE string(now) || ' Event indexReorg: ' || stmt
to console;
EXECUTE IMMEDIATE stmt ;
END LOOP indexLoop;
CLOSE indexLevels;
message string(now(*)) || 'Event indexReorg completed.'
to console;
END
go
ALTER EVENT "indexReorg"
    ADD SCHEDULE "indexReorg_1"
        START TIME '04:00:00' ON ('Sunday')
go
```

ディスクスペースの減少など、監視対象のサーバマシンに物理的にアクセスする必要があるアイテムについては、コンソールへのメッセージ送に加えてより迅速な注意喚起手段が必要です。これに関してSQL Anywhereは、MAPIインタフェースまたはSMTPサーバを介した電子メール送信機能を備えていません。

NOTE

Microsoft Exchangeのユーザーの場合は注意が必要です。MicrosoftはSPAMを減らす目的で、Outlook以外のクライアントからMAPIインタフェースへの電子メールの自動送出手続きを防止する仕組みを設けました。したがってMicrosoft Exchangeをご使用の場合は、無人サーバのアプリケーションにMAPIインタフェースを使わせるのは望ましくありません。このような環境では電子メールの送出手続きにSMTPを使う方がよいでしょう。

以下に示すのはメインのデータベースドライブの空きディスクスペースを検出するイベントの例です。データベースが複数のdbSpacesを使う場合は、データベースは各dbSpacesが参照する個々のドライブ名をチェックします。いずれかのドライブ上のディスクスペースが10%以下になると、イベントはdbOptionsテーブルに格納された管理アカウントに自動的に電子メールを送ります。

```
CREATE EVENT "lowDiskSpace"
    TYPE "DBDiskSpace"
    WHERE event_condition('DBFreePercent') <= 10
    ENABLE AT ALL
HANDLER
BEGIN
    declare sMailType char(4);
    declare sMailAcct varchar(60);
    declare sMailPwd varchar(60);
    declare sMailServer varchar(60);
    declare iSMTPPort smallint;
    declare sRecip varchar(60);
    declare SiteName varchar(60);
    declare iStatus integer;

    SELECT
        sMailType = mailType,
        sMailPwd = mailPwd,
        sMailServer = mailServer,
```

リモートサポートを考慮した データベースアプリケーション構築

```
iSMTPPort = mailPort,  
sRecip = adminEmail,  
SiteName = site_name  
FROM dbOptions ;  
  
CASE sMailType  
  WHEN 'MAPI' THEN  
    iStatus =  
      CALL xp_startMAIL(  
        mail_user = sMailAcct,  
        mail_password = sMailPwd );  
  
  WHEN 'SMTP' THEN  
    IStatus =  
      CALL xp_startSMTP(  
        smtp_sender = sMailAcct,  
        smtp_server = sMailServer,  
        smtp_port = iSMTPPort );  
  
END CASE;  
  
If iStatus <> 0 then  
  MESSAGE 'Low Disk Space Warning!! ' ||  
    'Cannot start Mail session. ' ||  
    'status = ' || string(iStatus)  
  to console;  
  return  
end if;  
  
iStatus =  
  CALL xp_sendMail(  
    recipient=adminEmail,  
    subject='Free disk space below 10% at: ' || SiteName  
  );  
If iStatus <> 0 then  
  MESSAGE 'Low Disk Space Warning!! ' ||  
    'Cannot send e-mail to: ' || adminEmail ||  
    ' via ' || sMailtype || '. status = ' ||  
    string(iStatus)  
  TO console ;  
end if;  
  
CALL xp_stopMail() ;  
END
```

データベースの診断

サポートコールが何度も入ってくるのに、「システムの速度が低下している」という情報しかないとき、問題が何であり、それがどこで起きているのかを正確に診断するのは管理者の仕事です。本章では、データベースエンジン内部の仕組みを明瞭に可視化し、パフォーマンスの低下に関連する一般的な要因を正確に把握するために実行できる簡単な操作と非侵入的な調査手段をご紹介します。

“Remember_Last_Statement”エンジンプロパティの設定

–zエンジンスイッチは、アクティブな接続が実行した最後の文をすべて「記憶」する機能を有効化します。この機能をオンにして放置してください。これにより各接続が使っている正確なSQL文を見ることができますが、発生する実行オーバーヘッドはごくわずかです。この機能を有効または無効にするには以下のコマンドを実行します。

```
call sa_server_option( 'remember_last_statement', 'ON' );
```

“Request_Level_Logging”の実行

Request_Level_Loggingは、データベースエンジンが処理するすべてのリクエストの包括的なトレースを提供するデバッグ機能で、パフォーマンスが低下しているクエリを正確に判断する際に役立ちます。

NOTE

この機能はデータベースエンジンに相当な負荷をもたらすので賢明に使わなくてはなりません。出力をファイル化すると、ファイルは急速に増大します。

この機能は以下のコマンドにより有効になります。

```
call sa_server_option( 'request_level_logging', 'SQL' );  
call sa_server_option( 'request_level_log_file', '<output file>' );
```

出力ファイルとして結果を取得した後は、以下のコマンドを実行してrequest_level_loggingをオフにしてください。

```
call sa_server_option( 'request_level_logging', 'none' );  
call sa_server_option( 'request_level_log_file', '' );
```

request_level_loggingが生成した出力ファイルを分析するには、以下の二つのストアードプロシージャを実行してください。

```
call sa_get_request_profile(  
    'request_level_log_file' [,connection-id ] );
```

sa_get_request_profile() は、生成されたログファイルを処理し、テンポラリテーブルSATMP_REQUEST_PROFILEに反映します。connection_id引数オプションを使うと、結果が特定のconnection_idにいたるまでフィルタリングされます。SATMP_REQUEST_PROFILEテーブルへの入力後、それをクエリすることでパフォーマンスが低いクエリを確認できます。

```
call sa_get_request_times(  
    'request_level_log_file' [,connection-id ] );
```

sa_get_request_times()は、sa_get_request_profile()と同様のプロシージャですが、こちらは処理したログファイルをsatmp_request_timesテーブルに反映します。

ブロックされた接続の検出

すでに見てきたとおり、SQL Anywhereはストアドプロシージャの結果を横断してSELECT操作を行う機能を備えています。この機能により、あらゆる変則的なコンディションを検出するクエリを記述することが可能です。

一例として、以下に別の接続にブロックされたすべての接続と、両方の接続が実行している文を検出するクエリを示します。

NOTE

このクエリは、“remember_last_statement”エンジンプロパティが有効になっているとほかに効果的です。

```
SELECT  
    blockedActivity.number                blockedConn,  
    blockedActivity.userID              blockedUser,  
    datediff( mi, blockedActivity.lastReqTime, now()) as blockedInMinutes,  
    blockedActivity.lastStatement      blockedStatement,  
    blockedInfo.nodeAddr                blockedIPAddr,  
    blockerActivity.number              blockingConn,  
    blockerActivity.userID              blockingUser,  
    datediff( mi, blockerActivity.lastReqTime, now()) blockingSince,  
    blockerActivity.lastStatement      blockingStatement,  
    blockerInfo.nodeAddr                blockingIPAddr  
  
FROM  
    sa_conn_activity() blockedActivity  
    JOIN  
        sa_conn_info() blockedInfo  
        ON blockedInfo.number = blockedActivity.number  
    JOIN  
        sa_db_info() dbInfo  
        ON blockedActivity.DBNumber = dbInfo.Number  
    JOIN  
        sa_conn_activity() blockerActivity  
        ON blockedInfo.blockedOn = blockerActivity.number  
    JOIN  
        sa_conn_info() blockerInfo  
        ON blockerActivity.number = blockerInfo.number  
ORDER BY  
    blockedInMinutes DESC  
;
```

ロングランクエリの検出

以下のクエリは、sa_conn_activity()およびsa_conn_info()プロシージャにより、SQL Anywhereエンジンが処理中のすべてのアクティブなクエリの現在の経過時間を報告するものです。

```
SELECT
```

```
Activity.number,  
Activity.userID,  
datediff( ms, Activity.lastReqTime, now()) / 1000.0 as duration,  
Activity.lastStatement,  
Info.nodeAddr IPAddr,  
Info.reqType requestType  
FROM  
sa_conn_activity() Activity  
JOIN  
sa_conn_info() Info  
ON Info.number = Activity.number  
WHERE  
RequestType not in ('COMMIT', 'CURSOR_CLOSE')  
ORDER BY  
Activity.lastReqTime;
```

エピソード

このクエリは最近VCIの顧客のサイトの一つでその効果を発揮しました。

CHUM TelevisionグループはCitrixサーバのクラスタ上にVCIのアプリケーションクライアントソフトウェアをホストしていましたが、ユーザーはパフォーマンスが低いと不満を表明していました。VCIでは上述の診断クエリを使ってロングランクエリを確認し、その大多数が「CURSOR_FETCH」というrequestTypeを持っている点に注目しました。これはデータベースエンジンがクエリの実行を完了し、結果セットをそのリクエスト(Citrixサーバにホストされたクライアントセッション)に返そうとしたことを示します。Citrixサーバを調べたところ、その処理能力が著しく不足している上、仕様より三倍も多くのクライアントセッションをサポートしようとしていたことがわかったのです。クラスタにCitrixサーバを追加することでクライアント側のボトルネックが軽減され、VCIのアプリケーションの性能は正常に戻りました。

データベースのローカル・コピーの取得

アクティブなユーザを持つ運用中のデータベースに対して実行できない診断試験は数多くあります。実行時にデータベースへの排他的なアクセスを必要とし、しかも完了まで一時間以上を要するsa_table_page_usage()ストアドプロシージャはその一例です。したがって場合によっては運用データベースのバックアップをラボに持ち帰ってローカルなサーバマシンに実装し、その上で診断試験を行うのが最善の選択肢となります。こうすればデータベースを思うままに止めたり再開したりできるし、Sybase Centralの統合デバッグでプロシージャ、トリガ、イベントごとに段階を踏んで試験を進めることもできます。また一般にデータベースへの排他的なアクセスを必要とするあらゆる試験を行うことができます。

SQL Anywhereのデータベースはプラットフォーム同士で完全にバイナリ互換なので、事前準備なしで各種のサーバに転送し実装することが可能です。またSQL AnywhereのDBファイルは圧縮性が高く、伝送前にWinZipやWinRarなどのユーティリティで.db及び.logファイルを圧縮できます。WinRar等のツールは、圧縮後のファイルサイズが4GBを越えたアーカイブを、CDへの焼き付けが可能な700MB大のファイルに自動的に分割できます。

一つのSQL Anywhereデータベースが物理的に複数のDBファイルと単独のLOGファイルから構成される可能性があります（複数のDBSPACEを使う場合）。イメージバックアップを行う場合は、データベースをローカルのサーバに実装する前に、これらのファイルをすべて転送しておく必要があります。また必要なデータベース構成要素をすべて含むアーカイブファイルを一つ生成するアーカイブバックアップも可能です。この場合はファイルを圧縮してローカルに移した後、一回のRESTORE DATABASE操作で個々のファイルを解凍できます。

まとめ

リモート組み込みアプリケーションのサポートは、多大な時間を要する困難な作業になりがちです。アプリケーション・スイートの開発に投じた労力と努力が、たった一つの運用データベースを失うことで無に帰してしまうこともあり得るのです。だからこそ開発段階からリモートで顧客をサポートする計画を立てておくことが大切です。繰り返しになりますが、計画では特に以下の事項に留意してください。

- サイトの準備
リモートサイトを注意深く計画し、十分に準備する。
- 定型業務の自動化
IT管理者がクライアントサイトに常駐しない場合でも定型業務が完遂できるような機能を組み込む。
- 問題の診断
運用データベースに起こった問題を診断するためのツールと技術を準備する。
- データベースのローカル・コピーの取得
問題を適切に診断して対策を講じるための試験用データベースをラボに構築する準備を整える。

ソフトウェアのサポートに関わるコストは単純に試算できないことがあります。ソフトウェア開発企業の収支決算に直接の影響を及ぼします。上述の四項目の実践はこうした見えないコストを最小限に抑え、顧客のTCOを低く維持するために重要な役割を演じるはずですが。

本書の著者について

本書の著者ポール・ホランはVideo Communications, Inc. (VCI) の上級設計者として、15年にわたってクライアント/サーバシステムと分散型アプリケーションの開発に携わってきました。リレーショナル・データベースの設計と実装を専門分野とするホランは、1993年以来Watcom / SQL Anywhereデータベースプラットフォームで開発を行っています。

ホランはiAnywhere Customer Advisory Panelの一員であり、また2000年からTeamSybaseのメンバーでもあります。毎年のSybase TechWave User Conferenceで多数の講演を行っています。現在ニューヨーク州バッファロー在住。

ホランのメールアドレス paulh@vcisolutions.com

法的注意

Copyright(C) 2006 iAnywhere Solutions, Inc. All rights reserved.

iAnywhere、iAnywhere Solutions、iAnywhere Solutions(ロゴ)、Adaptive Server、SQL AnywhereはiAnywhere Solutions, Inc.またはSybase, Inc.とその系列会社の米国または日本における登録商標または商標です。その他の商標はすべて各社に帰属します。

Mobile Linkの技術には、Certicom, Inc.より供給を受けたコンポーネントが含まれています。これらのコンポーネントは特許によって保護されています。

本書に記載された情報、助言、推奨、ソフトウェア、文書、データ、サービス、ロゴ、商標、図版、テキスト、写真、およびその他の資料(これらすべてを"資料"と総称する)は、iAnywhere Solutions, Inc.とその供給元に帰属し、著作権や商標の法律および国際条約によって保護されています。また、これらの資料はいずれも、iAnywhere Solutions, Inc.とその供給元の知的所有権の対象となるものであり、iAnywhere Solutions, Inc.とその供給元がこれらの権利のすべてを保有するものとします。

資料のいかなる部分も、iAnywhere Solutionsの知的所有権のライセンスを付与したり、既存のライセンス契約に修正を加えることを認めるものではないものとします。

資料は無保証で提供されるものであり、いかなる保証も行われません。iAnywhere Solutionsは、資料に関するすべての陳述と保証を明示的に拒否します。これには、商業性、特定の目的への整合性、非侵害性の黙示的な保証を無制限に含みます。

iAnywhere Solutionsは、資料自体の、または資料が依拠していると思われる内容、結果、正確性、適時性、完全性に関して、いかなる理由であろうと保証や陳述を行いません。Sybaseは、資料が途切れていないこと、誤りがないこと、いかなる欠陥も修正されていることに関して保証や陳述を行いません。ここでは、「iAnywhere Solutions」とは、iAnywhere Solutions, Inc.またはSybase, Inc.とその部門、子会社、継承者、および親会社と、その従業員、パートナー、社長、代理人、および代表者と、さらに資料を提供した第三者の情報元や提供者を表します。

リモートサポートを考慮した
データベースアプリケーション構築



アイエニウェア・ソリューションズ株式会社
<http://www.ianywhere.jp/>