6H-4

データベースアプリケーション プログラムの分散実行方式

二神 かほる　　池田 秀人
(広島大学総合情報処理センター)

1．はじめに

データベースの大規模化、業務システムの巨大化、多様化によりホストコンピュータにかかる負荷が非常に大きくなってきている。このため業務の遂行に支障をきたしている。これを解決する一つの方法としてアプリケーションプログラムの分散実行がある。これはホストで実行しているある部分をワークステーションに任せホストにかかる負荷を軽減させようとするものである。最近、ワークステーションの機能充実はめざましく、メニュー、グラフィクスなどワークステーションで実行する方がはるかに効率が上るものが多い。

分散実行するには、プログラムをホスト側とワークステーション側で実行する部分に分割しなければならない。ここではこの分割の一方法について述べる。

2．分割の基準と方法

広島大学では1980年10月よりデータベース管理システム・ＨＤＭのもとで、事務管理、図書館管理、病院情報管理のためのデータベースを構築してきた。この過程で全ての業務がＨＤＭのユーザ言語で記述出来たということから、この言語が実業務の手続きを記述する上で必要な能力を一応有していると考えられる。そこで言語はＨＤＭのユーザ言語を対象にした。分割の基本方針は次の５つである。

(1) ホスト側はデータベースに直接かかわる命令だけを実行する。
(2) 利用者がデータベース更新命令をワークステーション側から出してから、実際にデータベースが更新されるまでの時間的ずれを短かくする。
(3) データ転送量を出来るだけ少なくする。
(4) 障害時に分散実行による新しい問題が起こらないようにする。
(5) プログラムの分割が容易であり、分割されたものがプログラムとしてわかりやすい。

この方針のもとで言語をホストとワークステーションの機能にそれぞれ振り分けた。分割方法はワークステーションとホストが平行して動作する同期型を採った。実行形態を図１に示す。ホスト側で実行が必要なデータ操作コマンドは１０個である（表１）。図２はこれらのコマンドがホスト側で展開されるプログラムである。端末側で実行するプログラムは移行性を考慮してＰＡＳＣＡＬ言語を生成することにした。図３にユーザ言語プログラムと生成された端末側で実行するソースプログラムを示す。

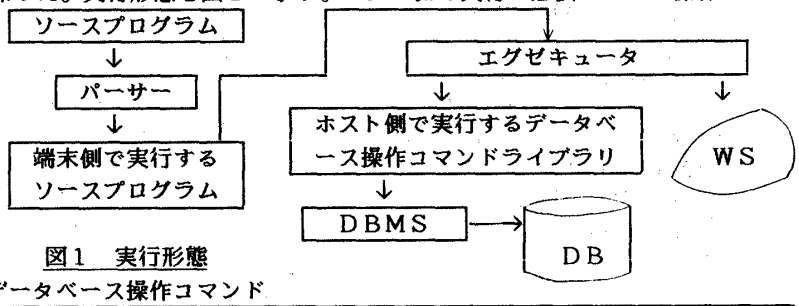

図１　実行形態

表１　データベース操作コマンド

コ　マ　ン　ド　名	機　能　概　要
OPEN　(ファイル名，パスワード，結果コード)	ファイルの使用を開始する。
CLOSE　(ファイル名，結果コード)	ファイルの使用を終わる。
FIND　(検索式，集合名，検索件数)	指定された条件を満足するレコードを検索する。
SET　(集合演算式，集合名)	集合演算をする。
USE　(集合名，項目名1:項目名2:...，レコード番号)	レコードを１件ずつ読みレコード番号も返す。
ADD　(R I ，項目名1=値1:項目名2=値2: ...，レコード番号)	レコードあるいは項目を追加する。
KILL　(R I ，項目名1=値1:項目名2=値2: ...，レコード番号)	レコードあるいは項目を削除する。
CHANGE (R I ，項目名1=値1:項目名2=値2: ...，レコード番号)	レコードあるいは項目を変更する。
SPRT　(集合名，ソート条件式，新集合名)	レコードを並べかえる。
COMMIT (制御式)	検索の競合、障害の対処をする。

Distributed execution method of Database application program

Kahoru FUTAGAMI, Hideto IKEDA
Hiroshima Univ.

```
FIND (検索式, 集合名, 検索件数)
  MORE
    集合名 :  検索式
    DUMMY1 :  COUNT RECORDS IN 集合名
       検索件数 = COUNT IN DUMMY1
  END MORE

USE (集合名, 項目名1:項目名2:..., レコード番号)
  MORE
    FOR EACH RECORD 集合名
       PRINT *RECORD          第1レコードのみ
    END FOR
  END MORE
```

```
MORE
  レコード番号 = ¥READ('')
  DUMMY2 : FIND ALL RECORDS 集合名 FOR WHICH
              POINT ¥ レコード番号 AND
              NOT POINT¥ レコード番号+1
           END FIND
           FOR 1 RECORD IN DUMMY2
              PRINT 項目1
              PRINT 項目2
                .
                .
           END FOR
           PRINT レコード番号
END MORE
```

図2　データ操作コマンドのホスト展開形

ソースプログラム

```
PROCEDURE HPRIORTY
OPENC DDDUSER
PASSWORD

BEGIN
    %CPRI IS STRING LEN 8 ARRAY(3)

    %CPRI(1) = 'HIGH'
    %CPRI(2) = 'STANDARD'
    %CPRI(3) = 'LOW'
    %UID = ¥READ('-- USER ID --')
L1: %PRI = ¥READ('-- PRIORITY --')
    %A   = ¥ONEOF(%PRI,'1/2/3','/')
    IF %A NE 1 THEN
       PRINT '** PRIORITY ERROR **'
       JUMP TO L1
    END IF

L2: IN DDDUSER FIND ALL RECORDS FOR WHICH
       CDUSER = %UID

L3: COUNT RECORDS IN L2
    IF COUNT IN L3 EQ 0 THEN
       PRINT '** UID(' WITH %UID WITH
             ') NOT REGISTERED **'
       JUMP TO L10
    END IF
    FOR 1 RECORD IN L2

       %PRV = LOGONPRV
       CHANGE PRIORITY TO %CPRI(%PRI)

       PRINT %UID AND %PRV
    END FOR
L10: *** CONTINUE
END
```

生成されたプログラム

```
PROGRAM HPRIORTY (INPUT,OUTPUT)
  LABEL L2;
  VAR CPRI : ARRAY [1..3] OF LSTRING(8);
  UID,PRI,PRV,LOGONPRV_ : LSTRING(20);
  A : INTEGER;
  I_1,IND_,L2RC_,L2RN_ : INTEGER;
  WRK_ : LSTRING(1024);
BEGIN
    HDMLOGON                                                    H
    OPEN ('DDDUSER','PASSWORD',IND_);
    IF IND_ <> 0 THEN GOTO L10;
      CPRI[1] := 'HIGH';
      CPRI[2] := 'STANDARD';
      CPRI[3] := 'LOW';
    UID := YREAD('-- USER ID --');
L1: PRI := YREAD('-- PRIORITY --');
    A   := YONEOF(PRI,'1/2/3','/');
    IF A <> 0 THEN
       BEGIN
         WRITELN ('** PRIORITY ERROR **');
         GOTO L1;
       END;
    CONCATH ('IN DDDUSER FIND ALL RECORDS FOR WHICH:
             CDUSER = ',UID,WRK_);
    FIND (WRK_,'L2',L2RC_);                                     H

    IF L2RC_ = 0 THEN
       BEGIN
         WRITELN ('** UID(',UID,') NOT REGISTERED **');
         GOTO L10;
       END;
    FOR I_1 := 1 TO 1 DO
      BEGIN
        USE ('L2','LOGONPRV',L2RN_);                            H
    PRV := LOGONPRV_;
    CONCATH ('PRIORITY=',CPRI[PRI],WRK_);
    CHANGE ('I',WRK_,L2RN_);                                    H
    WRITELN (UID,PRV);
      END;
L10: (** CONTINUE **);
END.
```

図3 ソースプログラムと生成されたプログラム　　H : ホスト側で実行