

Informix Dynamic Server 的抄寫技術

關於伺服器、網域與叢集

級別：中級

[Madison Pruet \(mpruet@us.ibm.com\)](mailto:mpruet@us.ibm.com)，IBM 資深技術人員

[Nagaraju Inturi \(nagaraju@us.ibm.com\)](mailto:nagaraju@us.ibm.com)，IBM 軟體工程師

[Nicholas Geib](#)，IBM 軟體工程師

2008 年 7 月 03 日

IBM® Informix® Dynamic Server (IDS) 是擁有強大抄寫技術的資料庫，其「企業技術 (ER)」與「高可用性資料抄寫 (HDR)」已為客戶服務多年，而新版的 IDS (11.1 版與目前的 11.5 版) 更為強大的資料庫增添新技術：RS 次要伺服器 (遠端獨立式次要伺服器) 與 SD 次要伺服器 (共用磁碟次要伺服器)。本文將簡介每種抄寫技術的特性，並針對其個別用法詳細說明基本觀念。

企業抄寫 (ER)

何謂 ER？

資料庫為業務儲存重要資料，不少業界人士都認為，將資料儲存在至少兩個不同的位置才是明智之舉。如果資料庫能夠自動將需要的資料抄寫至另一個伺服器，一定會更方便業務進行。全賴強大且靈活的「企業抄寫 (ER)」技術，IDS 7 版開始可以提供這項便捷的功能。您可以首先定義要抄寫資料的伺服器，藉此讓 ER 發揮作用，建立可傳輸資料的網路拓撲 (根節點、葉節點、非根節點與非葉節點)。ER 節點可能是單一伺服器，也可能是伺服器的叢集，本文將會在後續部分進一步說明。所有相互連結的節點合稱為 ER 網域，此網域只定義資料流動的路徑，不會定義傳輸的資料。

接下來，您要決定需要抄寫哪些資料，這部分是由 SQL 查詢所定義，其中涉及伺服器、資料庫、表格和欄位。此查詢可用來做為過濾器，其輸出會決定要抄寫的資料，且經過驗證，確實是具彈性的工具。如果要抄寫整個表格，查詢可以採用 `SELECT * FROM ...`；如果只要抄寫前三個直欄，您的查詢可以改成 `SELECT column1, column2, column3 FROM ...`；如果只抄寫特定列，您可以在查詢中使用 `WHERE` 子句。ER 可以使用很廣泛或精細的過濾器來抄寫資料，為了方便抄寫，ER 要求表格必須定義主要索引鍵。

查詢撰寫完畢之後，請決定哪些節點應參與資料抄寫。假設要在節點 A、B 與 C 的 `Employee` 表格顯示相同的資料，則 A、B 與 C 就是您的參與者 (*participant*)。您可以配置 ER，在伺服器之資料有所變更時，其他參與者也同時更新，這種狀況稱為到處更新 (*update anywhere*)。如果只要更新項目從 A 流動至 B 與 C，但不反向流動至 A 呢？ER 具有彈性的功能，讓您可以進行這種主要-目標 (*primary-target*) 狀況。一個以上的參與者資料均有變更時，伺服器 B 的列變更可能和伺服器 C 的列變更產生衝突；ER 可以讓您定義規則，自動處理這類衝突，包括比較時間戳記、執行儲存程序及忽略衝突。

瞭解查詢、參與者與不同狀況之後，您可以綜合這些資訊，以此建立抄寫。抄寫就像水龍頭一樣，控制來自查詢的資料流，這個水龍頭可自由開關，也可以有無限多的水龍頭。各種抄寫可以集結成抄寫集，如此，使用者即可更輕鬆地控制多重抄寫，此外，您還可以使用範本，為多種表格快速建立抄寫。這類操作均透過伺服器公用程式 `cdr` 執行。

建立及開始抄寫之後，資料實際上是如何移動的？首先，抄寫以非同步方式執行，這表示，資料在某個節點上確定之後，可能會有延遲，才會出現在另一個節點上；而且，只會抄寫確認資料。ER 會讀取邏輯日誌、測試邏輯記錄是否需要抄寫，並將資訊傳送給適當的參與者。

簡易範例

我們可以設定一個簡易的 ER 範例，在 `server1` 和 `server2` 之間，透過「到處更新」的方式抄寫「`db1`」資料庫的 `Employees` 表格 (下列步驟假設已使用 Informix 帳號)。

步驟 1：準備 SQLHOSTS 檔案

每部伺服器的檔案應包含兩部伺服器與兩個群組定義的登錄，每個群組代表一個 ER 節點。

grp1	group	-	-	i=1
server1	onsoctcp	host	port	g=grp1
grp2	group	-	-	i=2
server2	onsoctcp	host	port	g=grp2

步驟 2：準備 ER 磁碟空間

為每個伺服器執行下列步驟：

1. 編輯 ONCONFIG 以納入智慧型二進位大型物件空間：

```
CDR_QDATA_SBSPACES sbs1
```

2. 啟動伺服器。
3. 為二進位大型物件空間建立區塊，並且將空間新增至伺服器。例如，在 UNIX 上：

```
touch /ids/chunks/sbs1_chunk1  
chmod 660 /ids/chunks/sbs1_chunk1  
onspaces -c -S sbs1 -p /ids/chunks/sbs1_chunk1 -s 100000 -o 0
```

步驟 3：新增節點至新的 ER 網域

在 server1 執行下列指令：

```
cdr define server --init grp1
```

在 server2 執行下列指令：

```
cdr define server --sync=grp1 --init grp2
```

指令將建立兩個根節點的 ER 網域。

步驟 4：準備要抄寫的表格

在 server1 與 server2 執行下列 SQL 陳述式：

```
CREATE DATABASE db1 WITH LOG;  
CREATE TABLE employees(id int primary key, ...);
```

步驟 5：建立抄寫

在 server1 執行：

```
cdr define replicate --conflict=ignore r_employees ¥  
"dbl@grp1:informix.employees" "select * from employees" ¥  
"dbl@grp2:informix.employees" "select * from employees"
```

步驟 6：開始抄寫

在任一伺服器執行：

```
cdr start replicate r_employees
```

開始進行抄寫資料。若在 server1 或 server2 插入、更新或刪除列，另一個伺服器很快就會顯示。上述只是 ER 的基本操作，如需此技術與實用指令的相關資訊，請參閱 IDS InfoCenter 的「抄寫」小節（[參閱資源](#)）。

高可用性叢集

IDS 在 11.1 版之後，均具有叢集型抄寫功能。在叢集中，您可以取得橫向擴充、高可用性、災難回復與負載平衡功能，此技術已內建於資料庫，無須額外的硬體、軟體或網路基礎架構。叢集與 ER 不同，您無法在其中控制資料抄寫的精細度，並會抄寫整個伺服器的資料。叢集與 ER 技術可以並列使用，本文將在稍後再作說明。

叢集由一個主要伺服器及三種不同類型的次要伺服器所組成：

1. SD 次要伺服器（共用磁碟次要伺服器）
2. 高可用性資料抄寫 (HDR) 伺服器
3. RS 次要伺服器（遠端獨立式次要伺服器）

伺服器必須具相同的 IDS 版本，並且必須在相同的作業系統、硬體與磁碟架構上執行。在 11.1 版中，次要伺服器僅容許應用程式讀取資料；而在 11.5 版中，應用程式也可以插入、更新與刪除次要伺服器的資料，因此，所有三種次要伺服器類型都可以用來增加主要伺服器的容量，同時平衡負載。[表 1](#)（下方）比較伺服器類型。[表 2](#) 列出部分常見的 ONCONFIG 設定。

表 1：叢集伺服器類型的特性概觀

	主要伺服器	SD 次要伺服器	HDR	RS 次要伺服器
第一個 IDS 版本		11.1	6.00	11.1
主要用途		增加容量／橫向擴充	高可用性／快速待命	災難回復／遠端備份
每個叢集的節點數目	一個	不限	一個	不限
上限				
一個步驟中的可能類型變更	SD 次要伺服器、HDR、標準伺服器	主要伺服器	主要伺服器、RS 次要伺服器、標準伺服器	HDR、標準伺服器
支援的隔離等級	全部	錯誤讀取、已確定的讀取、已確定讀取	錯誤讀取	錯誤讀取

的前次確定

表 2：叢集伺服器類型常見的 ONCONFIG 參數

配置參數	伺服器類型	支援的值	說明
HA_ALIAS	SD 與 RS 次要伺服器	DBSERVERNAME 或其中一個 DBSERVERALIAS 名稱。必須是網路通訊協定。	用來定義名稱，以識別叢集中的次要伺服器。為您提供靈活彈性，可使用 DBSERVERNAME 之外的其他選擇。
LOG_INDEX_BUILDS	主要伺服器、HDR 與 RS 次要伺服器	0 - 停用索引頁面記載 1 - 啟用索引頁面記載	在主要伺服器上建立新索引時，「索引頁面記載 (IPL)」會在編輯日誌檔中寫入索引頁面。對 RSS 節點而言，必須啟用索引頁面記載，但 HDR 次要伺服器則可以自由選擇。
REDIRECTED_WRITES	HDR、SD 與 RS 次要伺服器	0 - 停用次要伺服器上的寫入功能 >=1 - 啟用次要伺服器的寫入功能	決定主要與次要伺服器之間，用以支援資料更新的網路連線數目。
TEMPTAB_NOLOG	主要伺服器、HDR、SD 與 RS 次要伺服器	0 - 建立暫時表格，並預設啟用記載功能 1 - 建立不含記載功能的暫時表格	在 HDR、RSS 與 SD 次要伺服器上必須設定為 1。針對以 CREATE TEMP TABLE 或 SELECT INTO TEMP 陳述式建立的明確暫時表格，控制其預設記載模式。次要伺服器不能記載暫時表格。

以下部分會進一步說明這些伺服器類型，探索其特性、ONCONFIG 參數，同時使用各種類型的伺服器來建置叢集。

SD 次要伺服器（共用磁碟次要伺服器）

SD 次要伺服器與主要伺服器共用磁碟空間（不包括暫存資料庫空間），共用方式通常是透過網路型叢集檔案系統。新增 SD 次要伺服器至叢集並不困難，只要備妥共用磁碟，幾秒鐘之內即可完成。因為 SD 次要伺服器節點使用主要伺服器的磁碟，而且可以輕鬆且快速擴充，很適合用於橫向擴充。在 SD 次要伺服器中，檢查點已經同步，這表示，唯有完成 SD 伺服器上的檢查點之後，主要伺服器的檢查點才會完成。從 IDS 11.5 版之後，SD 次要伺服器支援已確定的讀取、已確定讀取的前次確定，以及錯誤讀取的隔離等級。您可以單一指令，將 SD 次要伺服器提升為主要伺服器：`onmode -d make primary <name of SD server>`。SD 次要伺服器與主要伺服器有密切的關係（共用相同磁碟），因此，如果主要伺服器遇到問題，SD 次要伺服器最適合用來做為初步的失效接手。

表 3：重要的 SD 次要伺服器 ONCONFIG 參數

配置參數	伺服器類型	支援的值	說明
SDS_ENABLE	主要伺服器、SD 次要伺服器	0 - 停用 SDS 功能 1 - 啟用 SDS 功能	使用此指令將 SD 次要伺服器新增至叢集。
SDS_PAGING	SD 次要伺服器	<分頁檔 1 的絕對路徑>、<分頁檔 2 的絕對路徑>	必須配置兩個分頁檔以擴充 SDS 節點。

	伺服器	頁檔 2 的絕對路徑>	
SDS_TEMPDBS	SD 次要伺服器	<資料庫空間名稱>、<路徑>、 <頁面大小 (KB)>、<偏移 (KB)>、<大小 (KB)>	SD 次要伺服器節點的暫存資料庫空間資訊。您最多可以配置 16 項 SDS_TEMPDBS 登錄。 範例： SDS_TEMPDBS sdstmpdbs1, /work/dbspaces/sdstmpdbs1,2,0,16000
SDS_TIMEOUT	主要伺服器	>= 0 秒	此配置參數是供主要伺服器使用，以決定要耗費多久時間等待來自 SD 伺服器的認可。如果沒有認可，主要伺服器就會關閉 SD 伺服器。預設值為 20 秒。

新增 SD 次要伺服器至叢集

您可以採用獨立式 IDS 伺服器，將之變成叢集。首先，先新增 SD 次要伺服器（此實務情境假設已設定共用磁碟檔案系統，而且已使用 Informix 帳號）。

步驟 1：準備 SQLHOSTS 檔

確定在主要伺服器與 SDS 節點上的 SQLHOSTS 檔已互相擁有另一個伺服器上的登錄：

```
g_mach11 group - - i=10
myprim ontlitop primhost prim_port g=g_mach11
sds1 ontlitop sds1host sds1_port g=g_mach11
```

請注意，並未強制使用群組，不過，以下的範例有使用到群組。

步驟 2：將主要伺服器設定做為共用磁碟的擁有者

在主要伺服器上執行：

```
onmode -d set SDS primary myprim
```

步驟 3：配置 SD 次要伺服器

1. 確定以下項目符合主要伺服器的 ONCONFIG：ROOTNAME、ROOTPATH、ROOTOFFSET、ROOTSIZE、PHYSDBS、PHYSFILE、LOGFILES 與 LOGSIZE。
2. 將 SDS_ENABLE 設定為 1。
3. 配置 SDS_PAGING 與 SDS_TEMPDBS。

例如：

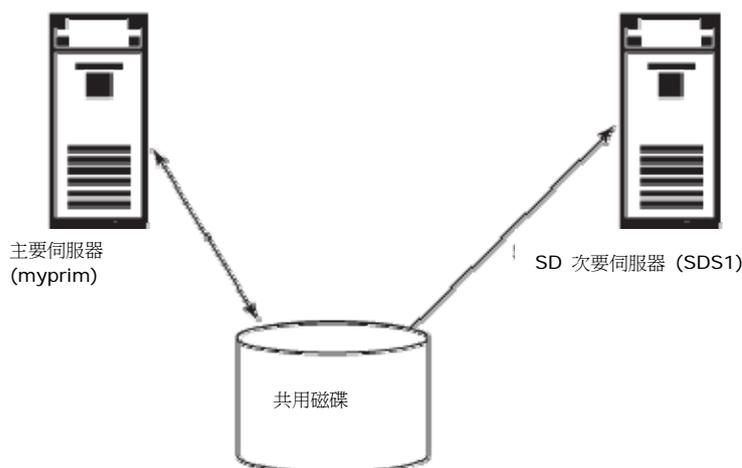
```
SDS_ENABLE 1
SDS_PAGING /ids/sds/dbspaces/page_1,/ids/sds/dbspaces/page_2
SDS_TEMPDBS sdstmpdbs1,/ids/sds/dbspaces/sdstmpdbs1,2,0,16000
REDIRECTED_WRITES 1
TEMPTAB_NOLOG 1
```

步驟 4：啟動 SD 次要伺服器

```
oninit
```

SD 次要伺服器已啟動並正在執行，您可以在主要與次要伺服器上，使用 `onstat -g sds` 指令來檢查叢集狀態。以下是叢集的結構：

圖 1. 叢集：主要伺服器與一個 SD 次要伺服器



SQLHOSTS 檔案詳細資料

```
g_match1f group - - i=10
myprim ondisc primhost prim_port g=g_match1f
sds1 ondisc sds1host sds1_port g=g_match1f
```

高可用性資料抄寫 (HDR)

高可用性資料抄寫或 HDR 是 IDS 中最早誕生的技術，由一對伺服器（主要伺服器和 HDR 次要伺服器）組成，同時支援同步與非同步抄寫模式。在同步模式中，主要伺服器需要收到 HDR 次要伺服器的認可，才會確定其交易。因此，HDR 次要伺服器可立即取代主要伺服器，即「快速待命」。在非同步模式中，主要伺服器與 HDR 次要伺服器之間只有檢查點會同步化。HDR 的其中一項特性是使用半雙工通訊協定，因此對於網路延遲非常敏感，可是，SD 與 RS 次要伺服器並沒有這項特性。

表 4：重要的 HDR 伺服器 ONCONFIG 參數

配置參數	伺服器類型	支援的值	說明
DRAUTO	主要伺服器、HDR 伺服器	0 - 手動 1 - 自動失效接手，最終保留伺服器類型 2 - 自動失效接手，最終逆轉伺服器類型 3 - 讓 Connection Manager	控制主要伺服器與 HDR 次要伺服器在故障時的行為。

Arbitrator 控制失效接手			
DRIDXAUTO	主要伺服器、	0 - 停用自動索引修復	如果 HDR 次要伺服器偵測到毀損，會自動執行索引修復。
	HDR 伺服器	1 - 啟用自動索引修復	
DRINTERVAL	主要伺服器	-1 - 採用同步模式操作	清除高可用性資料抄寫緩衝區的間隔時間上限（秒）。
		>= 0 - 採用非同步模式操作	
DRLOSTFOUND	主要伺服器、	<遺失協尋檔的路徑>	HDR 遺失協尋異動檔的路徑。此檔案是在失效接手期間建立，內含的交易已在主要伺服器上確定，但尚未在 HDR 伺服器確定。
	HDR 伺服器		
DRTIMEOUT	主要伺服器	>= 0 秒 預設值 = 30 秒	網路逾時的時間（秒），DRAUTO 以此來偵測故障。

新增 HDR 伺服器至叢集

在叢集中新增 HDR 次要伺服器節點。

步驟 1：準備 SQLHOSTS 檔案

更新主要伺服器、SDS 與 HDR 次要伺服器上的 SQLHOSTS 檔：

```
g_mach11 group - - i=10
myprim ontlitcp primhost prim_port g=g_mach11
hdr ontlitcp hdrhost hdr_port g=g_mach11
sds1 ontlitcp sds1host sds1_port g=g_mach11
```

步驟 2：配置 ONCONFIG 檔

為了讓 HDR 運作，特定的 ONCONFIG 參數必須與主要及次要伺服器的參數完全相同。在很多情況下，其中一個伺服器正在執行時，這些參數將無法變更。這些參數包括 DRAUTO、DRINTERVAL、DRTIMEOUT、根資料庫空間的相關設定，以及邏輯日誌的相關設定等等。為了確保這些設定值相符，其中一種做法是將主要伺服器的 onconfig 複製到次要伺服器，然後修改必須具有不同值的部分，諸如 DBSERVERNAME。在擴充主要伺服器之前，應該要規劃這些設定值，此範例使用的是預設值。

步驟 3：備份主要伺服器

在主要伺服器上執行 Level 0 備份：

```
ontape -s -L 0
```

步驟 4：向主要伺服器登錄 HDR 次要伺服器

在主要伺服器上執行：

```
onmode -d primary hdr
```

步驟 5：準備 HDR 次要伺服器的磁碟

在 HDR 次要伺服器上使用的儲存體必須與主要伺服器相符（例如，資料庫空間數目、資料塊數目、區塊大小、路徑名稱和偏移）。此範例使用備份來還原 HDR 次要伺服器，因此只需要具有區塊檔案，以及擁有適當的許可權即可。

步驟 6：還原 HDR 次要伺服器上的備份

在 HDR 伺服器執行 Level 0 備份的實體還原：

```
ontape -p

Three questions will be asked. Answer as shown below:

Continue restore? (y/n) y

Do you want to back up the logs? (y/n) n

Restore a level 1 archive (y/n) n
```

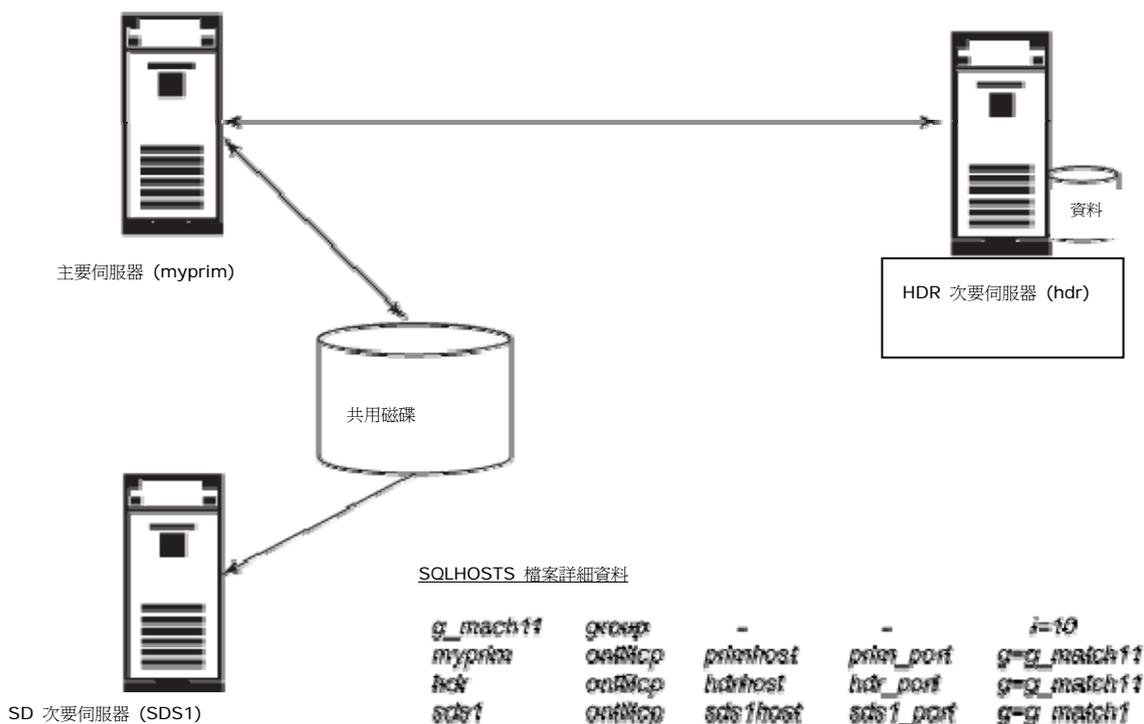
步驟 7：將 HDR 次要伺服器擴充為線上模式

完成還原之後，HDR 次要伺服器將進入回復模式。請執行下列指令：

```
onmode -d secondary myprim
```

HDR 次要伺服器已啟動，並正在執行。在主要伺服器與 HDR 次要伺服器上執行 `onstat -g dri`，以檢查 HDR 狀態。叢集目前的結構如下所示：

圖 2. 叢集：主要伺服器、一個 SD 次要伺服器與一個 HDR 次要伺服器



RS 次要伺服器（遠端獨立式次要伺服器）

RS 次要伺服器的主要用途為提供災難回復的解決方案。在 HDR 中，主要伺服器會持續將所有的邏輯日誌記錄傳送至 RS 次要伺服器，可是現在只採用非同步方式。可是，RS 次要伺服器與 HDR 不同，只使用全雙工通訊協定進行通訊，因此對網路延遲比較不敏感，而且可以常駐在遠距地點。RS 次要伺服器的其中一項特性是，主要伺服器與 RS 次要伺服器之間不會執行檢查點同步化，與 SD 和 HDR 伺服器不同。因此，RS 次

要伺服器無法立即取代主要伺服器，必須先變更為 HDR 伺服器。然而，若叢集發生多重故障，則 RS 節點有助於避免資料庫完全無法使用。

表 5：重要的 RS 次要伺服器 ONCONFIG 參數

配置參數	伺服器 類型	支援的值	說明
LOG_INDEX_BUILDS	主要伺 服器	0 - 停用索引 頁面記載 1 - 啓用索引 頁面記載	在主要伺服器上建立新索引時，「索引頁面記載 (IPL)」會在邏輯日誌檔中寫入索引頁面。主要伺服器上必須啓用索引頁面記載，才能新增 RS 伺服器至叢集。

新增 RS 次要伺服器至叢集

在叢集中新增 RS 節點。

步驟 1：準備 SQLHOSTS 檔案

叢集的所有伺服器必須擁有其他伺服器的 SQLHOSTS 登錄。

```
g_mach11 group - - i=10
myprim ontlitcp primhost prim_port g=g_mach11
hdr ontlitcp hdrhost hdr_port g=g_mach11
sds1 ontlitcp sds1host sds1_port g=g_mach11
rss1 ontlitcp rss1host rss1_port g=g_mach11
```

步驟 2：在主要伺服器上啓用索引頁面記載

```
onmode -wf LOG_INDEX_BUILDS=1
```

步驟 3：在主要伺服器上登錄新的 RS 次要伺服器

```
onmode -d add RSS rss1
```

步驟 4：執行主要伺服器 Level 0 備份

```
ontape -s -L 0
```

步驟 5：在 RS 次要伺服器上還原備份

```
ontape -p

Three questions will be asked. Answer as shown below:

Continue restore? (y/n) y

Do you want to back up the logs? (y/n) n
```

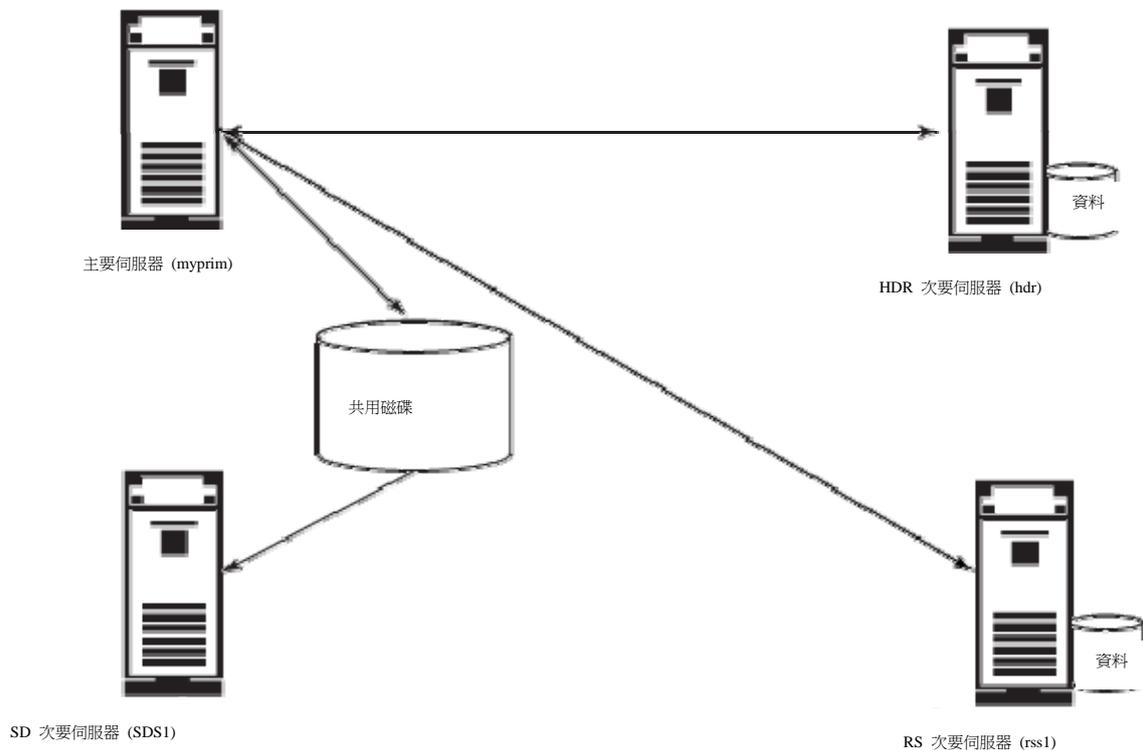
```
Restore a level 1 archive (y/n) n
```

步驟 6：將 RS 次要伺服器擴充為線上模式

```
onmode -d RSS myprim
```

RSS 節點已啟動，並正在執行。執行 `onstat -g rss` 指令以檢查主要伺服器與 RSS 節點的 RSS 節點狀態。叢集目前的結構如下所示：

圖 3. 叢集：主要伺服器、一個 SD 次要伺服器、一個 HDR 次要伺服器與一個 RS 次要伺服器



SQLHOSTS 檔案詳細資料

```
g_match11 group - - i=10
myprim ontcp primhost prim_port g=g_match11
hdr ontcp hdrhost hdr_port g=g_match11
sds1 ontcp sds1host sds1_port g=g_match11
rss1 ontcp rss1host rss1_port g=g_match11
```

總結

本文介紹了以 ER 執行非同步抄寫的彈性，以及叢集的高可用特性。兩者均可以搭配使用。之前曾經用於叢集 SQLHOSTS 檔的 "group"，也可以成為 ER 節點，就像在本文範例中的 server1 與 server2。您可以使用相同指令，新增叢集做為 ER 網域的節點，如同在獨立式伺服器上的做法：`cdr define server --sync=<sync node> --init <要新增的群組名稱>`。叢集為 ER 節點時，主要伺服器將執行與獨立式伺服器節點完全相同的職責；不過，由於主要伺服器是叢集，所以必須將資料抄寫至所有的次要伺服器。還有許多內容有待介紹，諸如個別技術的詳情、技術如何搭配運作以改善您的業務，以及連線管理程式 (Connection Manager) 與連續記錄還原 (Continuous Log Restore) 等相關技術的介紹，我們誠摯邀請您進一步瞭解 IDS！

資源

學習

- [IDS 11.5 資訊中心](#)：您可以在此處查看 *IDS* 文件。
- "[Availability Solutions with Informix Dynamic Server 11](#)" (IBM, 2007 年 5 月)：瞭解 *IDS 11* 的可用特性，這些特性可以視商業需要整合在一起，而且還能快速且輕鬆地進行配置及擴充。(PDF)
- "[What's new in IDS 11?](#)" (developerWorks, 2007 年 6 月)：參考 *IDS 11* 的新增特性，探索 *IDS* 的不同面向，例如可調整性、高可用性與效能、整合解決方案，以及免管理區域。
- [developerWorks Informix 專區](#)：尋找文章與指導教學，以及連接其他資源以拓展您的 *Informix* 技能。
- [developerWorks Information Management 專區](#)：進一步瞭解 *Information Management*，尋找技術文件、入門文章、教育、下載、產品資訊和更多內容。
- 透過 [developerWorks 技術活動與網路廣播](#)隨時獲得最新消息。
- [技術書店](#)：瀏覽上述與其他技術主題的相關書籍。

取得產品與技術

- [Informix Dynamic Server](#)：下載免費的 *IDS* 試用版。
- 使用 [IBM 試用軟體](#)來建置您的下一個開發專案，您可以從 *developerWorks* 直接下載。

討論

- [參加討論區](#)。
- [IDS Experts 部落格](#)：瀏覽由全球開發團隊與技術支援工程師所提供的 *Informix Dynamic Server* 技術附註。
- [Guy Bowerman 設立的 Informix Application Development 部落格](#)：尋找 *IDS* 應用程式開發的文章，並瀏覽作業系統、安全與其他主題。
- 參與 [developerWorks 部落格](#)，加入 *developerWorks* 社群。

關於作者

Madison Pruet 是 *IDS* 抄寫技術的架構工程師，從事 *IDS* 產品開發工作已有多年，目前負責 *IDS* 抄寫解決方案的架構設計，包括 *ER* 與 *HDR*。身為 *IDS 11* 開發的成員，他負責實作「連續可用性 (MACH11)」。

Nagaraju Inturi 加入 Informix Dynamic Server Replication 產品開發團隊已超過九年，是 IBM 認證的解決方案專家（Informix Dynamic Server 9 系統管理員）。

Nicholas Geib 是 IBM Lenexa 實驗室的軟體工程師，所屬之團隊負責支援 IDS 抄寫、備份與還原功能。