

Informix Dynamic Server
2008년 5월



IBM Information Management software

IDS 11.5 MACH11의 새로운 기능



Contents

1. MACH11 소개	1
2. High-Availability 클러스터를 구성하는 SQL Admin API 명령	2
3. High-Availability Secondary 서버에서 업데이트 지원	2
3.1 Secondary 서버 업데이트	2
3.2 Secondary 서버의 업데이트 가능 여부 탐지	3
3.3 쓰기 성능	4
4. High-Availability 클러스터용 Connection Manager	4
4.1 클라이언트 연결 관리	4
4.2 오류 해결 조정	7
4.3 타사 클러스터 관리자와의 통합	9
4.4 다중 이중화	9
4.5 연결 보안	10
4.6 Connection Manager 모니터링	11
5. OpenAdmin Tool과의 통합	12



[Introduction]

2007년 IBM은 가용성, 성능, 관리 등의 모든 데이터 관리 분야에서 모든 새로운 기능을 포함하는 IDS 11을 출시했습니다. 고급 데이터 관리를 위한 IDS 11의 다음 단계는 IDS 11.5입니다. IDS 11.5는 IDS의 모든 기존 장점과 더불어 더 많은 기능을 추가로 제공합니다. 본 문서는 MACH11에 관련된 새로운 기능을 살펴보는 IDS 11.5의 모든 새로운 기능에 대한 종합적인 백서입니다.

1. MACH11 소개

IDS Continuous Availability의 MACH11 개념은 IDS 11에서 처음 소개되었으며, 중요한 두 가지 새로운 복제 유형(SDS-Shared Disk Secondary 서버 및 RSS-Remote Standalone Secondary 서버)이 HDR 및 ER 모드에 추가되었습니다. 이러한 복제 유형이 함께 사용되어 IDS 서버의 다중 노드 클러스터를 구축하기 위한 다양한 구성요소를 제공합니다. 기존 HDR 복제와 마찬가지로 하나의 노드가 Primary 서버로 지정되지만 HDR과 달리 둘 이상의 Secondary 서버를 정의할 수 있습니다. 처리 용량을 늘리고 오류 해결 기능을 제공하기 위해 IDS 인스턴스의 클러스터를 사용할 수 있습니다.

SDS(Shared Disk Secondary) 서버

SDS 서버는 Primary 서버와 동일한 데이터를 공유 모드에서 사용하는 IDS 인스턴스입니다. 이 경우 여러 인스턴스가 같은 컴퓨터에 공존할 수 있으며, 모든 인스턴스가 기본 인스턴스와 동일한 데이터에 액세스합니다. 또한 컴퓨터 간에 디스크 장치를 공유할 수 있는 하드웨어 클러스터 환경에서 작동합니다. SDS 서버는 Primary 서버에 고장이 발생할 경우 Primary 서버의 역할을 수행할 수 있습니다.

RSS(Remote Standalone Secondary) 서버

RSS 서버는 데이터의 로컬 복사본을 유지한다는 점에서 HDR 인스턴스와 비슷합니다. RSS 서버는 Primary 서버와 비동기적으로 통신하며, HDR 동기 모드의 오버헤드가 너무 높은 경우에 사용됩니다. RSS 서버는 지리적으로 분산된 한 대 이상의 원격 Secondary 서버가 재해 복구 또는 고가용성을 위해 추가적인 워크로드 감소 및 가용성 향상 효과를 제공해야 하는 경우에 유용합니다.

HDR(High-Availability Data Replication) 서버

HDR 서버는 또한 MACH11 클러스터에 통합될 수 있으며, 고장 시 Primary 서버의 역할을 수행할 수 있는 Secondary 서버 기능을 제공합니다.

MACH11은 Primary 서버와 RSS 및 SDS 서버 간의 통신에 효율적인 전송 메커니즘을 사용하므로, 네트워크 대역폭이 제한된 상황에서 사용 가능합니다.

IDS 11 릴리스의 경우 MACH11 Secondary 서버가 “읽기전용(read-only)”이었지만, IDS 11.5 릴리스에서는 이러한 제한이 제거되었습니다. 해당 내용은 본 문서의 뒷부분에서 자세히 설명합니다.

2. High-Availability 클러스터를 구성하는 SQL Admin API 명령

이제 SQL Administration API를 사용하여 High-Availability 클러스터를 구성할 수 있습니다. API를 사용하여 동일한 `onmode -d` 명령을 수행하지만 구문이 약간 다릅니다. 연산 및 매개변수를 지정하기 위해 'HA command string', 'argument'를 사용합니다.

예를 들어, 하나의 인스턴스가 이미 정의되어 동일한 물리적 컴퓨터에서 SDS 서버 인스턴스를 사용하기 위해서는 다음 구문을 실행하십시오.

```
execute function task ('ha sds set', 'demo_ids_2');
```

전체 SQL Administration API 명령 세트는 IBM Informix Dynamic Server Guide to SQL: Syntax에 설명되어 있습니다.

3. High-Availability Secondary 서버에서 업데이트 지원

IDS 버전 11.5에서는 클라이언트 애플리케이션이 데이터 업데이트 트랜잭션을 Secondary 서버로 보낼 수 있도록 Secondary 서버를 구성할 수 있습니다.

3.1 Secondary 서버 업데이트

클라이언트 애플리케이션이 Secondary 서버를 업데이트가 가능한 서버로 설정하고자 한다면, `UPDATABLE_SECONDARY` 구성 매개변수의 값을 Primary 서버와 Secondary 서버 간의 SMX 파이프 수를 나타내는 양수값으로 설정하십시오. 이 매개변수의 권장 값은 CPU 가상 프로세서 수의 두 배입니다. Secondary 서버를 읽기 전용으로 만들려면 `UPDATABLE_SECONDARY`를 0으로 설정하십시오.

이 기능은 *IBM Informix Dynamic Server Administrator's Guide* 및 *IBM Informix Dynamic Server Administrator's Reference*에 설명되어 있습니다.

다음 예제에서 SDS 서버는 기본적으로 읽기 전용(read-only) 인스턴스입니다.

```
$ onstat -
IBM Informix Dynamic Server Version 11.50.UC1          -- Read-Only (SDS) -- Up
00:01:37 -- 166296 Kbytes
$ dbaccess stores_demo -
Database selected.
> begin;
Started transaction.
> update orders set ship_date = current where order_num = 1023;
346: Could not update a row in the table.
140: ISAM error: operation illegal on a DR Secondary
Error in line 1
Near character position 59
```

\$ONCONFIG에서 UPDATABLE_SECONDARY를 1로 설정하고 서버를 다시 시작하면 Secondary 서버에서도 업데이트 작업을 할 수 있습니다.

```
$ onstat -
IBM Informix Dynamic Server Version 11.50.UC1      -- Updatable (SDS) -- Up
00:00:11 -- 166296 Kbytes
$ dbaccess stores_demo -
Database selected.
> begin;
Started transaction.
> update orders set ship_date = current where order_num = 1023;
1 row(s) updated.
>
```

†UPDATABLE_SECONDARY 구성 매개변수를 이전에는 REDIRECTED_WRITES(IDS 11.50.xc1에서)라고 불렀습니다.

3.2 Secondary 서버의 업데이트 가능 여부 탐지

참고 : 클라이언트가 Updatable Secondary 서버에 연결한 경우, 클라이언트가 연결한 이 인스턴스가 Primary 서버가 아님을 나타내기 위해 sqlca.sqlwarn.sqlwarn6 플래그가 'W' 로 설정됩니다. 인스턴스의 쓰기 수락 여부를 감지하기 위해 이 방식을 사용했던 애플리케이션은 이에 따라 수정되어야 합니다.

DataBlade 프로그래머를 위해 mi_hdr_status() 함수가 개선되어 서버 상태를 나타내는 광범위한 비트맵을 반환합니다. 값은 \$INFORMIXDIR/incl/public/milib.h에 정의된 것처럼 다음과 같은 플래그 비트로 구성됩니다.

```
/*
 * HDR states.
 */

#define MI_HDR_ON                0x01
#define MI_HDR_PRIMARY          0x02

/*
 * Both MI_HDR_SECONDARY and MI_SECONDARY designate if the server is any kind
 * of secondary node. Prior to IDS 11.5 there was only one secondary type,
 * the HDR secondary.
 */

#define MI_HDR_SECONDARY        0x04 /* for backwards compatibility */
#define MI_SECONDARY            0x04 /* recommended over MI_HDR_SECONDARY */

/*
 * MI_HDR_SEC_NODE designates if the server is exactly and only an HDR
 * secondary node (and not another secondary type node).
 */

#define MI_HDR_SEC_NODE         0x08
#define MI_RSS_SECONDARY        0x10
#define MI_SDS_SECONDARY        0x20
#define MI_UPDATABLE_SECONDARY  0x40
```

onstat - 를 사용하고 해당 명령의 반환 코드를 검사하는 경우, 읽기 전용의 업데이트 가능 서버는 Secondary 서버임을 나타내기 위해 상태 2를 반환합니다. 서버 상태에 대한 자세한 정보는 onstat -g sds 및 onstat -g rss를 사용하여 확인할 수 있습니다.

```

IBM Informix Dynamic Server Version 11.50.UC1          -- On-Line -- Up 00:04:42 --
166296 Kbytes

Local server type: Primary
Number of SDS servers:2

SDS server information

SDS srv      SDS srv      Connection      Last LPG sent      Supports
name         status       status           (log id,page)      Proxy Writes
db_sds_ro    Active       Connected        57,1395             N
db_sds_rw    Active       Connected        57,1395             Y
    
```

Proxy Writes 열은 Secondary 서버에 쓰기가 가능한지의 여부를 나타냅니다.

3.3 쓰기 성능

Secondary 서버는 다른 모든 쿼리와 마찬가지로 Insert/Update/Delete 쿼리를 처리합니다. 그러나 Secondary 서버에서 물리적 페이지에 쓰기를 수행해야 하는 시점에 도달하면 이 서버는 대신 Primary 노드에 쓰기 요청을 보냅니다. 쿼리의 대부분을 Secondary 서버에서 처리하므로 Primary 서버에 미치는 영향은 굉장히 줄어듭니다. 즉, Primary 서버와 Secondary 서버 사이에 차이점을 생각할 필요가 없으므로 SDS 서버의 클러스터를 관리하는 것이 매우 쉽습니다. 이것은 Connection Manager Arbitrator를 검사할 때 더욱 명확하게 알 수 있을 것입니다.

또한 데이터 행의 버전 관리를 선택적으로 구현하기 위해 새로운 기능이 추가되었습니다. 이렇게 하면 전체 물리적 페이지를 비교하지 않고도 데이터 행에서 primary 소스와 secondary 소스 간의 차이점을 서버가 신속하게 파악할 수 있습니다.

4. High-Availability 클러스터용 Connection Manager

새로운 Connection Manager는 High-Availability 클러스터에 있는 가장 적절한 서버로 클라이언트 애플리케이션 연결 요청을 동적으로 전달합니다. Connection Manager는 클러스터에 있는 각 서버에 연결되어 서버 유형, 미사용 워크로드 용량 및 서버의 현재 상태에 대한 통계를 수집합니다. 이러한 정보를 바탕으로 Connection Manager는 클라이언트 연결을 적절한 서버로 리디렉션합니다. 또한 Connection Manager Arbitrator는 자동 오류 해결(failover) 로직을 High-Availability 클러스터에 제공합니다.

4.1 클라이언트 연결 관리

Connection Manager는 IDS 인스턴스를 논리적으로 그룹화할 수 있습니다. 각 논리적 그룹을 SLA(Service Level Agreement; 서비스 수준 계약)라고 하며, SLA에는 INFORMIXSERVER 이름 또는 서버 유형 (SDS/RSS/HDR 등)별로 서버 인스턴스의 내역이 포함됩니다.

다음의 클러스터 설정에는 오류 해결 및 추가적인 확장성을 제공하기 위한 두 대의 SDS 서버가 있습니다. 이 때 클라이언트가 어떤 서버에 연결할 것인지를 어떻게 알 수 있을까요? 기존 메커니즘에서는 서버를 sqlhosts 파일에 그룹으로 구성하지만, 여기서는 확장성과 오류 해결 시 높은 가용성을 위해 기본적인 '라운드 로빈' 방식의 연결만 제공합니다.

다음은 클라이언트 인터페이스로 작동하여 세 가지 SLA를 제공하는 Connection Manager를 나타냅니다.

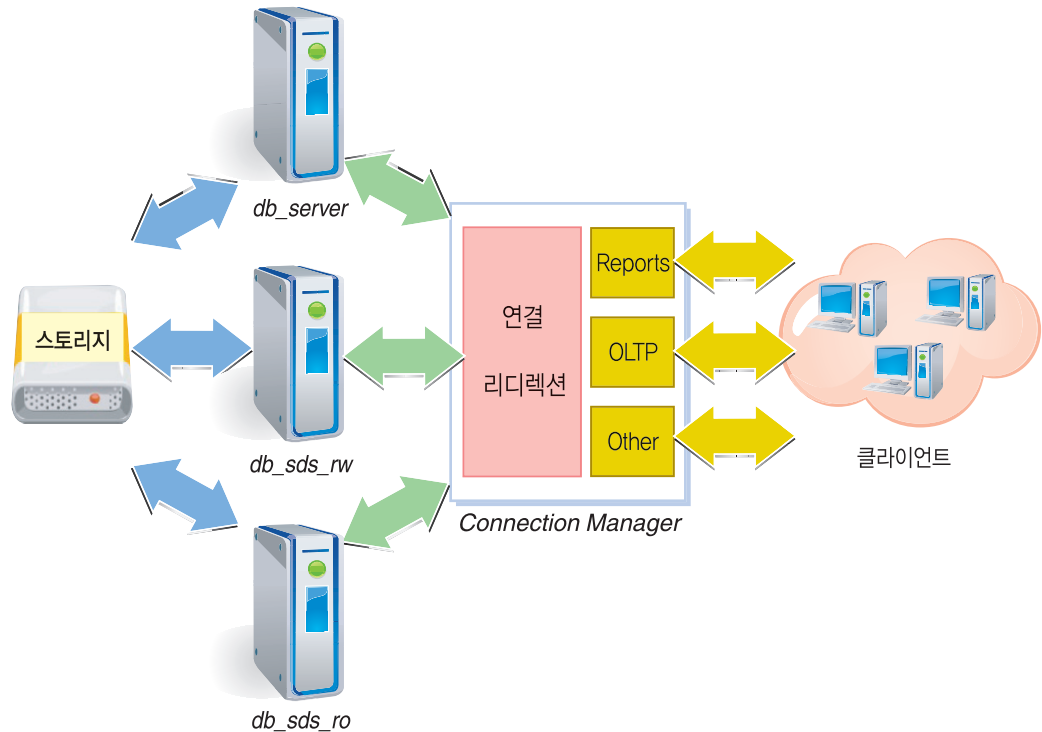


그림 1. 세 가지 SLA가 있는 Connection Manager 클라이언트 인터페이스

다른 유형의 클라이언트에 필요한 서버 특성을 반영하도록 다음 서비스 수준이 선택되었습니다.

Reports - 대용량 쿼리 실행에 최적으로 구성된 인스턴스로서, 서버를 분리할 만큼 데이터 가용성이 그다지 중요하지 않은 경우의 기본적인 읽기 전용 서비스입니다. 그림 1에서는 읽기 전용 SDS 서버인 db_sdzs_ro입니다.

OLTP - 고성능의 업데이트 중심 서비스입니다. 그림 1에서는 Primary 서버인 db_server입니다.

Other - 위의 두 가지 경우의 중간 정도 수준이며, 업데이트는 허용하지만 동시성이 다소 떨어질 수 있습니다. 그림 1에서는 업데이트가 가능한 SDS 서버인 db_sds_rw입니다.

각 클라이언트는 필요에 따라 위의 서비스 유형을 제공하는 서버에 선택하여 연결할 수 있지만, 이 경우 서버의 명명된 인스턴스가 "하드 코딩(hard coding)"됩니다. 서버가 사용 불가능하거나 클러스터의 또 다른 서버가 온라인 상태가 되어 동일한 서비스를 제공하는 경우, 수동으로 개입하여 클라이언트 구성을 변경하지 않는 한 클라이언트는 이 상황을 알 수 없습니다. 또한 각 클라이언트는 로드 밸런싱을 수행하고 모든 서버 상의 부하를 기준으로 최적의 서비스 수준을 제공할 수 있도록 연결하는 것도 좋습니다.

Connection Manager는 위와 같은 특성으로 정의된 서비스를 제공함으로써 이 문제를 해결하며, 해당 서비스를 서버 인스턴스 그룹에 매핑합니다. 따라서 위의 세 가지 서비스 유형은 다음과 같이 수행합니다.

- Reports - db_sds_ro에 연결하며, 사용할 수 없는 경우 db_sds_rw에 연결합니다.
- OLTP - db_server 또는 db_sds_rw 중 로드가 적은 쪽에 연결합니다.
- Other - 처음에는 db_sds_rw에 연결하지만 사용할 수 없는 경우 db_server에 연결합니다.

이 정보는 명령행을 통해 명령어로 Connection Manager에 전달될 수 있지만(oncmsm) 그룹 정보를 구성 파일에 저장하는 것이 더 유용합니다. 기본적으로 이 파일은 \$INFORMIXDIR/etc/cmsm.cfg이지만 oncmsm -c filename을 사용하여 명령행에서 지정할 수도 있습니다. Connection Manager의 작업에 대한 광범위한 로그가 생성되므로, 로그 파일의 위치를 -과 함께 명령행에 지정하거나 구성 파일에 위치를 정할 수 있습니다. 이 클러스터의 구성 파일은 다음과 같습니다.

```

NAME          ids_db
SLA           reports=db_sds_ro+db_sds_rw
SLA           oltp=(db_server+db_sds_rw)
SLA           other=db_sds_rw+db_server
LOGFILE      /products/11.50/demo/server/oncmsm.log
    
```

"name=server_a+server_b+server_c"의 간단한 SLA 사양은 사용할 서버의 순서를 나타냅니다(server_a, server_b, server_c 순). "name=(server_a+server_b+server_c)"와 같이 괄호가 사용된 경우에는 모든 서버의 순위가 동일하며 로드가 가장 적은 서버에 우선순위가 부여됨을 나타냅니다. 또한 "name=(server_a+server_b)+server_c"와 같이 이러한 사양을 혼합할 수도 있습니다. 이 경우 로드를 기준으로 처음 두 서버가 동일한 순위로 고려되며, 두 서버를 모두 사용할 수 없는 경우에만 server_c가 고려됩니다.

서버의 로드는 서버를 모니터링하고 서버 간의 연결 대기 시간, 준비된 스레드 수, 서버의 세션 생성률 등에 대한 정보를 수집하여 계산합니다.

실제 서버 이름을 지정할 뿐만 아니라 SLA를 설정할 때 서버 유형만 사용할 수도 있습니다. 그러면 관리자가 일일이 이름을 지정하지 않아도 Connection Manager가 해당 서버 유형을 실제 dbserver에 매핑합니다. 서버 유형은 다음과 같습니다.

- PRI(primary) - 클러스터에서 Primary 서버로 지정된 서버
- SDS - SDS(공유 디스크 보조) 서버
- HDR - 기존 HDR(High-Availability Data Replication) Secondary 서버
- RSS - RSS(원격 독립형 보조) 서버

위와 같은 서버 정의에서 SLA 보고서를 reports=SDS라고 지정할 수도 있습니다.

명시적 서버 이름과 서버 유형을 조합할 수는 있지만, 목록에서 연결 후보를 검색할 경우에는 각 서버가 한 번만 나타납니다.

Connection Manager는 일반적인 SQL 클라이언트에 대한 연결 지점으로 작동하므로, 자체적으로 sqlhosts 구성 파일에 입력된 값이 있어야 합니다.

```
# Primary server
db_server  onsocp      koetsu.lan  9088
db_drda    drsocp      koetsu.lan  9089

# SDS servers
db_sds_ro  onsocp      koetsu.lan  9098
db_sds_rw  onsocp      koetsu.lan  9108

# Connection Manager services
oltp       onsocp      koetsu.lan  19188
reports    onsocp      koetsu.lan  19189
other      onsocp      koetsu.lan  19190
```

클라이언트는 연결 사양의 INFORMIXSERVER 부분을 reports, oltp, other 중 하나로 설정합니다.

Connection Manager는 초기 연결이 요청될 때만 관여합니다. Connection Manager는 선택한 서버로 연결을 넘겨주고 클라이언트와 서버 간의 데이터 흐름에 더 이상 관여하지 않습니다. 클라이언트가 연결되는 실제 서버는 DBSERVERNAME 값을 쿼리하는 기존의 방식으로 찾을 수 있습니다.

```
SELECT DBSERVERNAME FROM TABLE(SET{1});
```

연결을 리디렉션하는 메커니즘에서는 하위 레벨 클라이언트-서버 연결 프로토콜의 기능을 사용하며, 클라이언트는 Client SDK 버전 3.50 이상을 사용해야 합니다.

4.2 오류 해결 조정

Connection Manager의 또 다른 역할은 클러스터 환경에서 서버 오류 해결을 처리하는 것입니다. 이 작업은 명령행에 -f 플래그를 지정하거나 구성 파일에 FOC(Fail Over Configuration) 매개변수를 사용하여 수행됩니다. 사양은 서버 및 서버 유형(SLA와 유사하지만 PRI/Primary 서버 유형 생략)의 순서 목록과 타임아웃 값으로 구성됩니다.

괄호 안에 입력된 항목인 경우, 특정 서버 별칭 이름의 순위가 더 높으며 SDS, HDR, RSS와 같은 일반적인 서버 유형 이름 순으로 순위가 정해집니다.

Connection Manager가 Primary 서버에 고장이 발생한 것을 탐지한 경우, 이후 타임아웃 중에 Primary 서버가 재연결 작업을 수행하지 않으면 FOC 목록에 나열된 그 다음 서버 또는 서버 유형이 Primary 서버의 후보로 선택됩니다. 그러나 네트워크 통신 장애이지만 Connection Manager가 Primary 서버에 고장이 발생한 것으로 잘못 진단하지 않도록 하려면 먼저 Secondary 서버에서도 Primary 서버 고장이 탐지되었는지 확인해야 합니다.

기본 FOC는 "SDS+HDR+RSS,0"이며, 이것은 Primary 서버에 고장이 발생한 경우 SDS 서버가 Primary 서버를 대체하는 첫 번째 후보가 된다는 의미입니다. Connection Manager Arbitrator는 타임아웃 시간(이 경우 0 초) 동안 기다린 다음 SDS 서버를 Primary 서버로 전환하는 데 필요한 명령을 생성합니다. SDS 서버를 사용할 수 없는 경우에는 사용 가능한 HDR 서버, RSS 서버 순으로 선택되어 Primary 서버로 전환됩니다.

Connection Manager Arbitrator를 해제하려면 FOC DISABLED를 사용하십시오.

클러스터의 모든 노드가 동일하게 간주되는 정상적인 상황에서는 일반적인 서버 유형(예: SDS)을 사용하는 것이 유용합니다. 한 서버에 고장이 발생하고 또 다른 서버가 Primary 서버로 승격되도록 기다리는 경우 수행할 수 있는 유일한 관리 작업은 고장이 발생한 노드를 또 다른 SDS 서버로 다시 시작하는 것입니다. 이러한 이유로 Primary 서버는 필요한 onconfig 매개변수(SDS_PAGING, SDS_TEMPDBS 및 SDS_ENABLE)를 이미 정의했어야 합니다. 다음 예는 Connection Manager Arbitrator가 수행하는 이벤트 순서입니다.

```
Wed Mar 19 16:34:12 2008
16:34:12 IBM Informix Connection Manager
16:34:12 Connection Manager name is ids_db
16:34:12 Starting Connection Manager...
16:34:12 Current max open fd is 1024
16:34:12 switch to daemon mode
16:34:12 Connection Manager for ids_db is already running
16:34:12 dbservername      = db_server
16:34:12 nettype                = onsoctcp
16:34:12 hostname                = 10.0.0.103
16:34:12 servicename           = 9088
16:34:12 options                = s=0
16:34:12 listener reports initializing
16:34:12 listener other initializing
16:34:12 listener oltp initializing
16:34:12 Listener other=(db_sds_rw+db_server) is active with 8 worker threads
16:34:12 Listener reports=db_sds_ro+db_sds_rw+primary is active with 8 worker threads
16:34:12 Listener oltp=db_server+db_sds_rw is active with 8 worker threads
16:34:12 Connection Manager successfully connected to db_server
16:34:13 Arbitrator FOC string = db_server+SDS,0
16:34:13 FOC[0] = db_server
16:34:13 FOC[1] = SDS
16:34:13 FOC timeout = 0
16:34:13 Arbitrator FOC string = db_server+SDS,0
16:34:13 FOC[0] = db_server
16:34:13 FOC[1] = SDS
16:34:13 FOC timeout = 0
16:34:13 Connection Manager successfully connected to db_sds_ro
16:34:13 Connection Manager successfully connected to db_sds_rw
16:34:13 Arbitrator is active on CM = ids_db
16:34:13 Connection Manager started successfully
```

위의 메시지는 Connection Manager가 초기화되었으며 클러스터의 모든 서버와 연결할 수 있었음을 나타냅니다. 이제 경고 없이 Primary 서버 db_server가 종료됩니다.

```
16:38:29 Connection Manager disconnected from db_server
16:38:35 Arbitrator FOC string = db_server+SDS,0
16:38:35 FOC[0] = db_server16:38:35 FOC[1] = SDS
16:38:35 FOC timeout = 0
16:38:41 Arbitrator make primary on node = db_sds_ro successful
```

SDS 서버 중 하나인 db_sds_ro가 Primary 서버로 자동 승격됩니다.

최초 Primary 서버인 db_server의 문제가 수정되면, Primary 서버가 SDS 서버로 되돌아오게 되어 클러스터를 다시 구성할 수 있습니다.

```
16:39:48 Connection Manager successfully connected to db_server
```

경우에 따라 최초 Primary 서버로 다시 돌아가기 위해, 알려져 있는 고정된 Primary 서버 이름과 오류 해결 시퀀스를 보유하는 것이 바람직할 수 있습니다. 명명된 Primary 서버가 최초 서버가 되도록 FOC를 설정하십시오. 예를 들어, 다음 예제에서와 같이 FOC db_server+SDS,0인 경우 모든 오류 해결 상황에서 항상 Primary 서버와 동일한 서버로 다시 돌아갑니다. 새로 승격된 Primary 서버가 종료되고, 이전의 Primary 서버가 원래 역할을 이어받습니다.

```
16:40:48 Connection Manager disconnected from db_sds_ro
16:40:56 Arbitrator FOC string = db_server+SDS,0
16:40:56 FOC[0] = db_server
16:40:56 FOC[1] = SDS
16:40:56 FOC timeout = 0
16:41:01 Arbitrator make primary on node = db_server successful
16:45:41 Connection Manager successfully connected to db_sds_ro
```

마지막으로, db_sds_ro(새로운 Primary 서버였던 노드)를 다시 시작한 후 일반 SDS 서버로 클러스터에 다시 배치할 수 있습니다.

4.3 타사 클러스터 관리자와의 통합

하드웨어 클러스터 환경에 따라 오류 해결 시 운영 체제의 클러스터 관리 소프트웨어가 특정 작업을 수행해야 합니다. 이것은 디스크 서브시스템을 다시 구성해야 하는 경우에 가장 중요합니다. 새로운 Primary 서버를 시작할 수 있도록 읽기 전용 공유 디스크를 쓰기 가능 디스크로 전환하는 작업을 I/O Fencing이라고 합니다.

O/S 제공 클러스터 관리 소프트웨어를 Connection Manager에 통합하기 위해 새로운 구성 매개변수인 FAILOVER_CALLBACK이 사용되었습니다. 이 매개변수는 Secondary 서버가 Primary 서버나 표준 서버로 승격될 때마다 호출되어야 하는 스크립트를 지정합니다. 이 스크립트는 Secondary 서버의 상태가 변경되기 전에 호출되며, Primary 서버가 되는 서버 상의 공유 디스크에 대한 쓰기 액세스를 허용하고 고장이 발생한 서버의 공유 디스크에 대한 쓰기 액세스를 해제합니다.

전환 중에 스크립트가 문제를 감지하면 Non-zero 오류 코드와 함께 스크립트가 종료되어야 합니다. 그러면 심각한 문제가 발생했다는 신호가 Connection Manager에 전달되고 DBA 관리 상태에서 수동으로 전환을 수행해야 합니다.

이 기능은 IBM Informix Dynamic Server Administrator's Guide에 설명되어 있습니다.

4.4 다중 이중화

Connection Manager를 보유하고 있다는 것이 고가용성 시나리오에서 매우 중요하지만, Connection Manager 자체가 고장 지점이 되어서는 안 됩니다. 그렇다면 고장을 완벽하게 방지하려면 어떻게 해야 할까요?

먼저, Connection Manager를 임의의 서버 인스턴스에서 로컬로 실행할 필요가 없으며, 서버 컴퓨터에 발생한 고장이 영향을 미치지 않도록 Connection Manager를 다른 물리적 컴퓨터에서 실행할 수 있습니다. 연결 호스트 이름과 포트 번호를 나타내도록 sqlhosts 파일을 올바르게 구성하십시오. 또한 클러스터를 구성하는 IDS 인스턴스와 달리 Connection Manager는 서버 플랫폼이 아닌 다른 하드웨어 플랫폼에서 실행될 수 있습니다.

가용성을 늘리기 위해 여러 개의 Connection Manager를 실행하고 sqlhosts에 있는 서버 그룹의 기본 오류 해결 기능을 사용할 수 있습니다. oltp 서비스를 예로 들어 보겠습니다.

```
# Client connection configuration for oltp service
oltp      group      -          -          e=oltp_cm2
oltp_cm1  onsoctcp    cm1_host   9298      g=oltp
oltp_cm2  onsoctcp    cm2_host   9298      g=oltp
```

또한 이 그룹에 Primary 서버를 추가하면 두 Connection Manager에 고장이 발생하더라도 계속해서 연결 상태를 유지할 수 있습니다.

4.5 연결 보안

복잡한 네트워크 환경에서 여러 대의 서버를 관리하기 위해서는 Connection Manager가 클러스터 내의 서버에 연결하고, 내부 서버 통계를 모니터링하며, 서버 간의 연결을 처리하고, 오류 해결 상황에서 서버를 관리할 수 있어야 합니다. 이러한 권한 레벨은 Informix 사용자에게만 부여되며, 클러스터의 모든 컴퓨터에 대해 이러한 권한 액세스 레벨을 허용하는 메커니즘이 있어야 합니다.

이러한 메커니즘을 수행하기 위해 클러스터의 서버에 필요한 사용자 및 암호 조합이 암호화된 파일에 저장됩니다. 암호화된 파일은 입력 파일로부터 생성됩니다. 입력 파일은 각 서버/사용자 조합에 사용되는 행 하나로 구성되어 있으며, 각 행은 dbservername, 대체 dbserveralias, 사용자 이름 및 암호와 같은 네 가지 필드로 구성되어 있습니다.

onpassword 유틸리티는 암호화된 파일을 생성하는 데 사용됩니다. 소규모 클러스터 환경의 입력 파일 (password.in)은 다음과 같습니다.

```
db_server  -      informix      password1
db_sds_rw  -      informix      password2
db_sds_ro  -      informix      password3
```

이 서버에는 대체 dbserveraliases가 없으므로, 두 번째 필드는 해당사항이 없으며 '-'는 단순히 자리 표시자로 사용됩니다. 사용자 Informix로 서버에 연결하려면 다른 암호가 필요합니다. 암호화된 파일 \$INFORMIXDIR/etc/passwd_file이 다음과 같이 생성됩니다.

```
onpassword -k encryption_key -e password.in
```

안전한 암호 저장 메커니즘을 사용할 뿐만 아니라 SSL을 사용하여 클라이언트, 서버 및 Connection Manager 간의 모든 연결을 암호화할 수 있습니다.

4.6 Connection Manager 모니터링

Connection Manager의 현재 상태가 로그 파일에 기록됩니다. 구성 파일에서 DEBUG 매개변수를 1로 설정하여 메시지의 수를 늘릴 수 있습니다. 기본 설정은 0입니다.

Connection Manager에서 생성된 로그 파일 외에도 Connection Manager의 상태 모니터링에 사용될 수 있는 새로운 onstat -g cmsm 옵션이 있습니다.

```
IBM Informix Dynamic Server Version 11.50.UC1      -- On-Line -- Up 1 days
00:29:38 -- 166296 Kbytes

CM name  host      sla      define          foc          flag  connections
ids_db   koetsu   reports  db_sds_ro+db_sd db_server+db_sds  3      0
ids_db   koetsu   oltp     db_server+db_sd  db_server+db_sds  3      135
ids_db   koetsu   other    (db_sds_rw+db_s db_server+db_sds  3      30
```

SLA 및 FOC 내역은 onstat의 단순한 출력 스타일에 맞추기 위해 잘리지만, SMI 테이블을 통해 전체 내역을 볼 수 있습니다.

```
> select * from sysmaster:syscmsm;

sid          265
name         ids_db
host         koetsu
foc          db_server+db_sds_rw+db_sds_ro,0
flag         3
sla_id       0
sla_name     reports
sla_define   db_sds_ro+db_sds_rw+primary
connections  0

sid          265
name         ids_db
host         koetsu
foc          db_server+db_sds_rw+db_sds_ro,0
flag         3
sla_id       1
sla_name     oltp
sla_define   db_server+db_sds_rw
connections  135

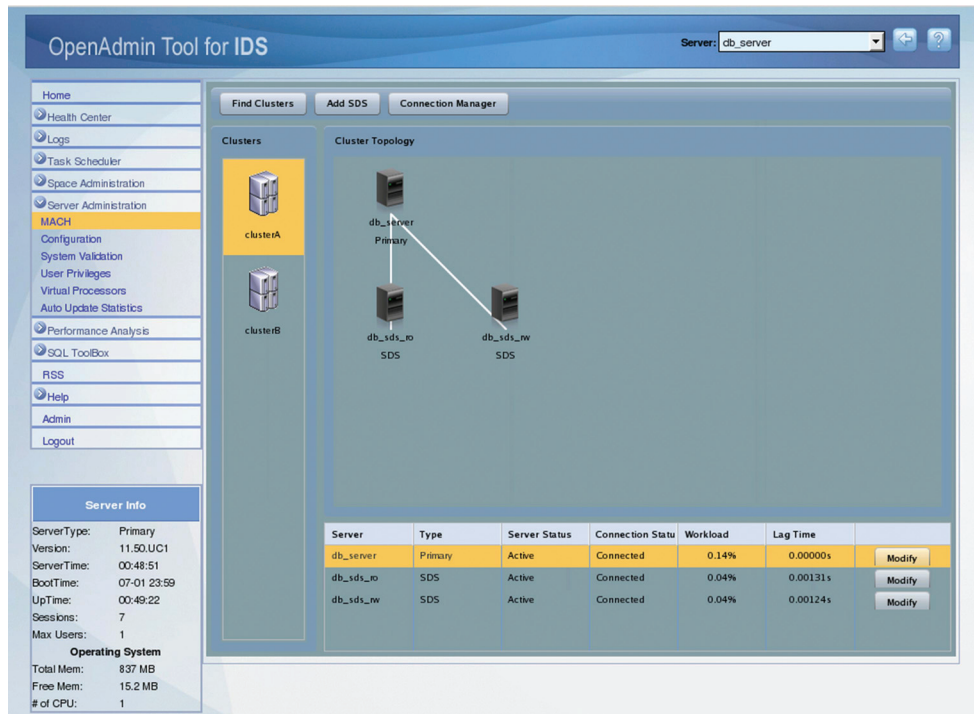
sid          265
name         ids_db
host         koetsu
foc          db_server+db_sds_rw+db_sds_ro,0
flag         3
sla_id       2
sla_name     other
sla_define   (db_sds_rw+db_server)
connections  30

3 row(s) retrieved.
```

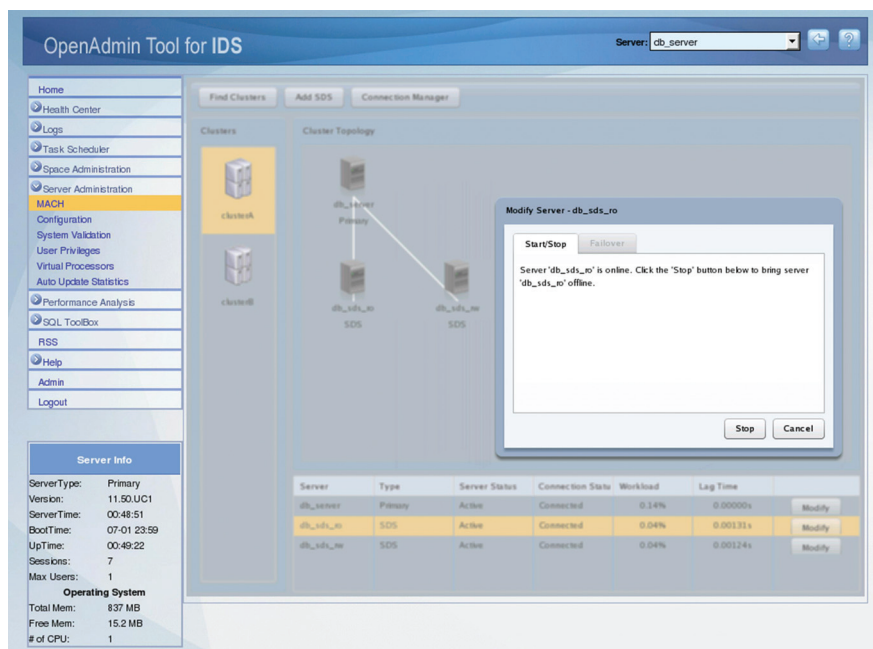
Connection Manager는 Client SDK와 함께 설치되며, 대부분 IBM Informix Dynamic Server Administrator's Guide 및 IBM Informix Dynamic Server Administrator's Reference에 설명되어 있습니다.

5. OpenAdmin Tool과의 통합

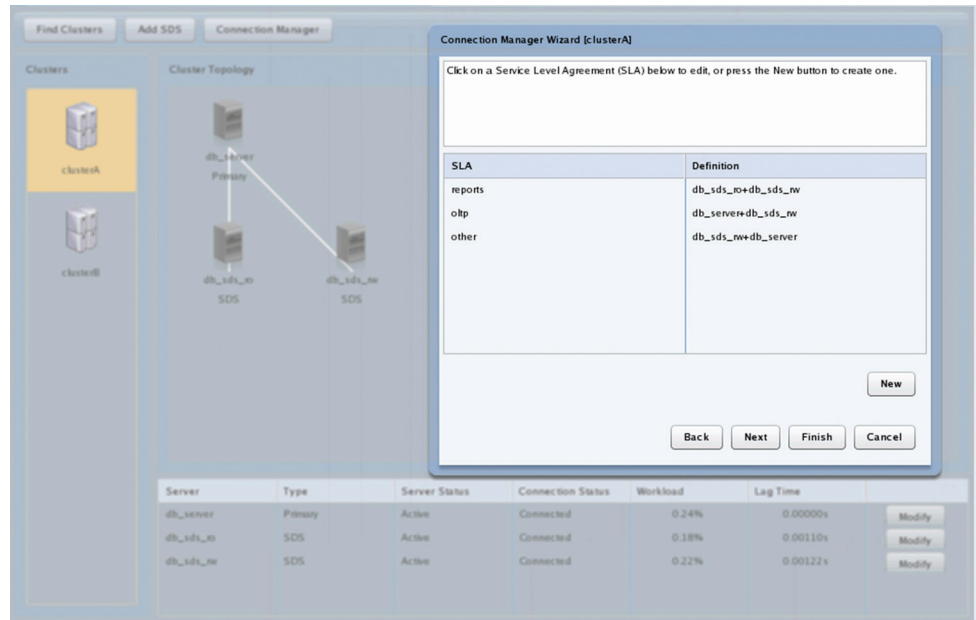
OAT(OpenAdmin Tool)의 기능 향상을 통해 MACH11 클러스터 환경의 모니터링과 관리가 더욱 단순해집니다. 클러스터의 서버에 연결할 때 MACH 관리 화면에서 Find Clusters를 클릭하면 전체 클러스터 토폴로지를 매우 쉽게 볼 수 있습니다.



Modify 단추를 사용하면 클러스터 내 각 노드의 상태를 변경할 수 있습니다.



또한 OAT를 사용하여 Connection Manager를 관리할 수 있습니다.



OAT의 새로운 기능은 OAT Release Notes에 설명되어 있습니다.

사용된 참조 문서

- IBM Informix Dynamic Server Guide to SQL: Syntax, Version 11.50
- IBM Informix Dynamic Server Administrator's Guide, Version 11.50
- IBM Informix Dynamic Server Administrator's Reference, Version 11.50
- OAT Release Notes – <http://localhost:8080/openadmin/README.html>, Version 2.20



© Copyright IBM Corporation 2008

(135-270) 서울시 강남구 도곡동 467-12
군인공제회관빌딩

한국아이비엠주식회사
고객만족센터

TEL: (02)3781-7114
www.ibm.com/kr

2008년 7월

Printed in Korea
All Rights Reserved