

SQL Server 2012 – AlwaysOn 을 통해 본 SQL Server 업그레이드 필요성 (Mirroring과 비교)

(주)엑셈 컨설팅본부/SQL Server팀 양 동환

AlwaysOn의 등장

2012 년 상반기 MicroSoft 에서 Big Data, Cloud Computing 시장을 겨냥해 SQL Server 2012 를 출시하면서 New Feature 로 'AlwaysOn'이라는 획기적인 HA 기술이 추가되었다. 이전 버전들에서 HA 방식으로 지원되던 Failover Clustering 이나 Mirroring 등의 단점들을 보완하기에 충분해 보인다. 본 내용은 AlwaysOn 을 통해 SQL Server 하위버전을 쓰는 고객들에게 왜 SQLServer2012 로 엔진을 업그레이드 해야 되는지 그 필요성을 설명하고 있다. 그럼 먼저 AlwaysOn 에 대해 간략히 설명하겠다.

AlwaysOn Availability Group(가용성 그룹)이란?

'AlwaysOn 가용성 그룹' 기능은 데이터베이스 Mirroring 에 대한 엔터프라이즈 수준의 대안을 제공하는 고 가용성(HA) 및 재해 복구(DR) 솔루션이다. 가용성 그룹은 여러 사용자 데이터베이스를 그룹 단위로 묶어서 장애 조치(failover)할 수 있는 환경을 지원한다. 가용성 그룹은 읽기/쓰기가 가능한 주 데이터베이스 복제 노드와 보조 데이터베이스 복제 노드(최대 4 개까지)를 지원한다. 필요한 경우 보조 데이터베이스 복제 노드에 대해 읽기 전용 액세스를 설정하거나 일부 백업 작업에 사용되도록 설정할 수 있다. 가용성 그룹은 가용성 복제본의 수준에서 장애 조치 된다. 따라서 데이터 파일 손실, 데이터베이스 삭제, 트랜잭션 로그 손상 등으로 인해 주의 대상 데이터 베이스가 발생할 경우의 문제로는 장애 조치가 수행되지 않으므로 유의해야 한다.

참조 : MSDN

그렇다면 알아보기 쉽게 이전버전의 HA 기능들과 비교해서 표를 통해 설명하겠다.

SQL Server 2008 R2		
기능	특징	단점
Failover Clustering	<ul style="list-style-type: none"> ☞ Instance 수준 고 가용성 ☞ Hot Standby ☞ DB 수준 고 가용성은 별도로 고려해야 함 ☞ 장애발생시 1분 내외 주/보조 서버 전환 가능 ☞ 많은 사례 확보 	<ul style="list-style-type: none"> ☞ Passive 로 구성된 서버 자원의 낭비 ☞ DB 파일에 대한 이중화 구성 필요 ☞ DR 에 대한 고민 필요 ☞ SAN(or DAS) 구성 필요 ☞ 다수의 클러스터 구성 시 각 클러스터 노드 통합 관제 어려움
Database Mirroring	<ul style="list-style-type: none"> ☞ DB 수준 고 가용성 제공 ☞ Hot Standby ☞ 30 초 내외 자동 장애 복구 ☞ 주/보조(Mirror) 서버간의 2 Phase Commit 으로 완벽한 데이터 동기화 제공 	<ul style="list-style-type: none"> ☞ 1:1 제한으로 복수 보조 데이터베이스 구성 불가능 ☞ 고 가용성을 위한 Connection String 에 Failover Partner 추가 ☞ 보조(Mirror)서버 데이터 활용을 위해서 DB Snapshot 별도 설정 ☞ DB 그룹으로 설정하여 주/보조 서비스 전환의 어려움
Replication	<ul style="list-style-type: none"> ☞ DB(Article 단위) 복제 ☞ Worm Standby ☞ 1:N DB 구성하여 Scale-out 가능 ☞ 보조 서버에서 조회작업 가능 ☞ 보조 서버에서 데이터 수정 가능 	<ul style="list-style-type: none"> ☞ 주 서버 장애 시 수동으로 서비스를 전환 하기 때문에 즉각적인 서비스 전환 불가능 ☞ 각 서버 노드에 대한 상태 모니터링 어려움 ☞ 복제 단위가 DB 가 아닌 개체(Object) 단위가 기 때문에 여러 개체 복제 시 까다로움
Log Shipping	<ul style="list-style-type: none"> ☞ DB 수준 DR 구성 ☞ Worm Standby ☞ 1:N 의 DB 구성하여 Scale-out 가능 ☞ 보조 서버에서 조회 작업 가능 	<ul style="list-style-type: none"> ☞ 주 서버 장애 시 수동으로 서비스를 전환 하기 때문에 즉각적인 서비스 전환 불가능 ☞ 주 서버와 보조 서버간 동기화 시간차 발생 ☞ 각 서버 노드에 대한 상태 모니터링 어려움 ☞ 로그 복구 시 모든 세션 연결 끊김 현상 발생

표 1. SQL Server 2008 R2 의 HA솔루션들

SQL Server 2012		
기능	비교 대상	특장점
AlwaysOn Availability Group	Failover Clustering	<ul style="list-style-type: none"> ☞ 보조서버 DB 에 대한 읽기 작업 가능 ☞ DB 파일에 대한 이중화 구성 제공 ☞ HA 와 DR 을 동시에 만족하는 솔루션 ☞ SAN(or DAS)와 같은 별도 장비 없이 구성 가능 ☞ AlwaysOn 대시보드를 통해 전체 노드에 대한 통합 관제 제공
	Database Mirroring	<ul style="list-style-type: none"> ☞ 최대 4 대의 보조 서버를 구성 가능 ☞ Listener 를 이용하여 단일한 Connection String 제공 ☞ 보조서버는 주 서버와 실시간으로 데이터를 동기화 하며 성능을 성능을 고려해 비 동기 모드도 지원 ☞ AlwaysOn Group 에 다양한 DB 를 그룹으로 묶어서 서비스 전환 가능 ☞ AlwaysOn 대시보드를 통해 전체 노드에 대한 통합 관제 제공
	Replication	<ul style="list-style-type: none"> ☞ 주 서버 장애 발생시 즉각적인 서비스 전환 제공 ☞ 주/보조 DB 서버에 대한 Connection 투명성 제공 ☞ AlwaysOn 대시보드를 통해 전체 노드에 대한 통합 관제 제공
	Log Shipping	<ul style="list-style-type: none"> ☞ 주 서버 장애 발생시 즉각적인 서비스 전환 ☞ 주/보조 DB 서버에 대한 Connection 투명성 제공 ☞ 주/보조 서버간의 실시간 데이터 동기화로 서비스 부하 분산 ☞ AlwaysOn 대시보드를 통해 전체 노드에 대한 통합 관제 제공

표 2. SQL Server 2012 의 HA 솔루션(AlwaysOn)

설명과 표를 통한 비교자료를 가지고도 AlwaysOn 의 장점을 이해하기 어렵고 필요성이 느껴지지 않는다고 생각된다면 다음 내용을 살펴보도록 하자. 다음은 2 가지의 장애 시나리오와 1 가지의 시나리오를 가지고 AlwaysOn 환경과 Mirroring 환경에서 어떤 차이가 나타나는지 비교해서 보여주도록 하겠다.

가용성 그룹 단위 Mirroring VS 단일 DB 단위 Mirroring

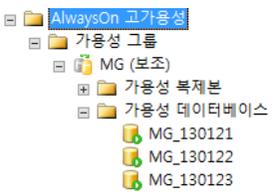
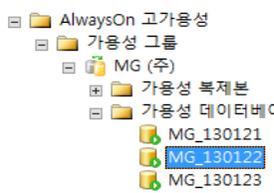
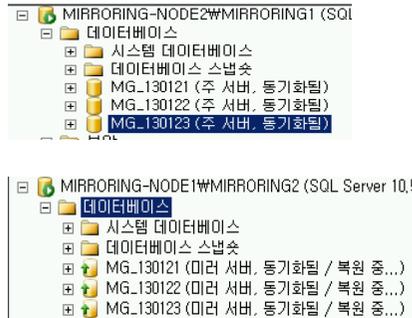
환경	AlwaysOn	Mirroring
OS Version	Windows Server 2012 EE	Windows Server 2012 EE
SQL Server Version	SQL Server 2012 EE	SQL Server 2008 R2 EE
서버 개수	3 대(AD, NODE1, NODE2)	3 대(NODE1, NODE2, Witness)
Mirroring DB 이름 및 개수	<p>3 개(가용성 그룹 1 개)</p> <p>MG(가용성 그룹)</p>  	<p>3 개(각각)</p> <p>MG_130121</p> <p>MG_130122</p> <p>MG_130123</p> 

표3 시나리오 1번 환경

각 환경에 구성된 DB 는 일자 별 Log 정보를 담고 있는 DB 들이다. (130121~130123) 이 DB 들 중에 1 개의 DB(mg_130122)에서 장애가 발생하였을 경우 3 개의 DB 모두의 정보를 필요로 하는 쿼리가 제대로 동작하는지 확인해 보았다

```

SELECT * FROM MG_130121.dbo.DB_FILE_SIZE_INFO
WHERE DB_NAME='BCPDB'
UNION ALL
SELECT * FROM MG_130122.dbo.DB_FILE_SIZE_INFO
WHERE DB_NAME='BCPDB'
UNION ALL
SELECT * FROM MG_130123.dbo.DB_FILE_SIZE_INFO
WHERE DB_NAME='BCPDB'
GO

```

100 % <

결과 메시지

	LOGTIME	DB_NAME	FILE_NAME	FILE_ID	FULL_FILE_NAME
1	2013-01-21 20:03:00.000	bcpDB	bcpDB	1	C:\Program Files\Microsoft
2	2013-01-21 20:03:00.000	bcpDB	bcpDB_log	2	C:\Program Files\Microsoft
3	2013-01-22 21:14:12.000	bcpDB	bcpDB	1	C:\Program Files\Microsoft

AlwaysOn 환경 쿼리 결과

```

USE master
GO
SELECT * FROM MG_130121.dbo.DB_FILE_INFO
WHERE DB_NAME='AdventureWorks'
UNION All
SELECT * FROM MG_130122.dbo.DB_FILE_INFO
WHERE DB_NAME='AdventureWorks'
UNION All
SELECT * FROM MG_130123.dbo.DB_FILE_INFO
WHERE DB_NAME='AdventureWorks'
go

```

메시지 954, 수준 14, 상태 1, 줄 2
 데이터베이스 "MG_130122"을(를) 열 수 없습니다. 미리 데이터베이스로 작동되고 있습니다.

Mirroring 환경 쿼리 결과

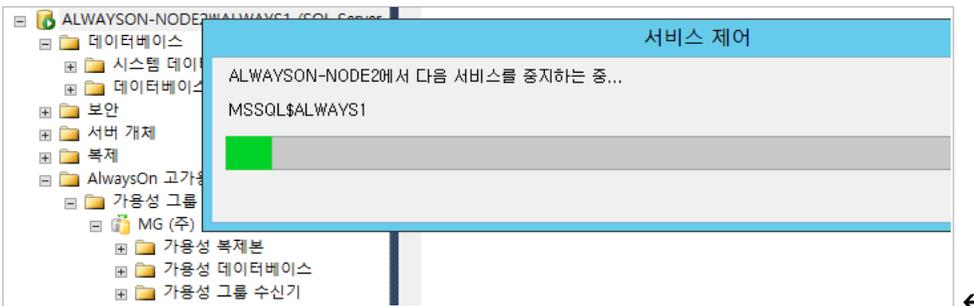
Mirroring 은 단일 DB 간 Mirroring 시스템이기 때문에 3 개의 DB 중에 하나가 failover 되면 쿼리가 실패를 하게 된다. 그에 반해 AlwaysOn 은 관련 있는 DB 들을 그룹으로 지정해서 그룹 단위로 Mirror 가 되기 때문에 위와 같은 경우 3 개 중 하나의 DB 라도 장애가 발생하면 그룹 채로 failover 되어서 쿼리가 실패할 경우가 발생하지 않는다.

최대 4노드 복제(동적 쿼럼 방식) VS 1:1 복제(과반수 방식)

환경	AlwaysOn	Mirroring
OS Version	Windows Server 2012 EE	Windows Server 2012 EE
SQLServer Version	SQL Server 2012 EE	SQL Server 2008 R2 EE
서버 개수	4대(AD,NODE1,NODE2, NODE3)	3대(NODE1,NODE2, Witness)
Mirroring DB 개수	3개(가용성 그룹 1개)	3개(각각)

표4 시나리오 2번 환경

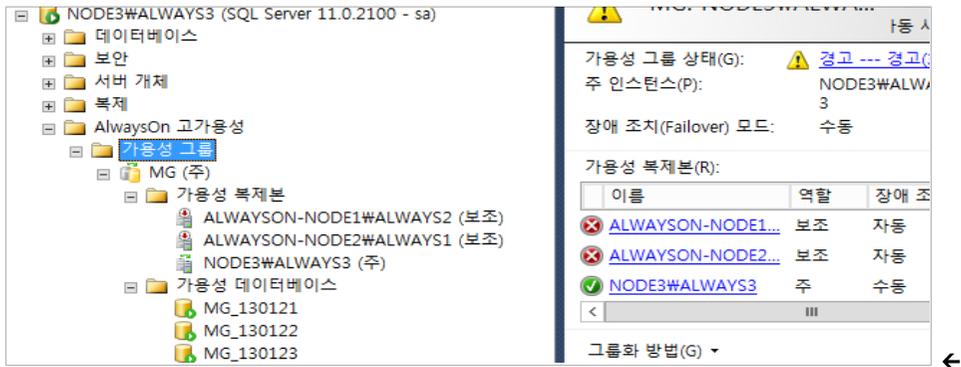
각 환경에 구성된 DB 역시 1 번 시나리오와 같은 DB 들이다. 여기서 서버들을 하나씩 장애를 발생시켜 보았을 때 어떻게 되는지 살펴보겠다.



NODE1 정지



NODE2 정지



NODE3으로 장애조치

이 시나리오를 통해 보여주고자 하는 내용은 Mirroring 은 1:1 로만 복제가 가능하고 AlwaysOn 은 최대 4 노드까지 복제가 가능하다는 점이다. 각 환경에서 서버에 장애가 발생할 경우 AlwaysOn 은 최대 4 대의 서버까지는 장애가 발생해도 사용이 가능하다는 점이고 Mirroring 은 2 대가 발생하면 더 이상 사용이 불가하다. 그 2 대가 DB 가 저장된 서버이든 모니터링을 위한 Witness 서버이든 상관없이 최대 3 대중 2 대만 Fail 이 일어나도 과반수 정책에 의해 작동하지 않게 된다. 만약 그 2 대가 같은 곳에 있는 서버들인데 그 곳에 정전이라도 발생한다면 어떻게 되겠는가? 생각만 해도 끔찍한 상황이 발생할 것이다. 물론 AlwaysOn 또한 자동 장애 조치가 가능한 것은 최대 2 대까지 이고 그 외에는 수동으로 장애 조치를 해야 한다. 그렇지만 최악의 경우가 발생했을 때 수동으로라도 장애 조치를 할 수 있는 것과 아무 손도 쓸 수 없는 것과는 천지차이다.

복제 노드 읽기 작업 및 백업 가능 VS. 복제 노드 스냅샷 한정 읽기 가능

환경	AlwaysOn	Mirroring
OS Version	Windows Server 2012 EE	Windows Server 2012 EE
SQLServer Version	SQL Server 2012 EE	SQL Server 2008 R2 EE
서버 개수	3대(AD, NODE1, NODE2, NODE3)	3대(NODE1, NODE2, Witness)
Mirroring DB 개수	3개(가용성 그룹 1개)	3개(각각)

표5 시나리오 3번 환경

마지막 시나리오는 보조 복제 노드의 활용이다. 단순 Mirroring 환경에서 보조 복제 노드는 스냅샷을 통한 한정된 읽기 작업밖에 가능한 것이 없다. 서버가 2 대이지만 1 대의 역할밖에 수행할 수 없는 비효율이 발생한다. 그렇지만 AlwaysOn 환경에서는 주 노드를 제외한 모든 복제 노드들에서 읽기 작업 및 백업 작업이 가능하다. 만약 AlwaysOn 환경에 보조 복제 노드가 2 개가 존재한다면 하나의 보조 복제 노드에서는 백업 작업을 실시하고 다른 보조 복제 노드에서는 보고서 작업과 같은 분석 업무를 실시할 수 있다. 또한 읽기 전용 라우팅(참조 :

[http://msdn.microsoft.com/ko-kr/library/7bd89ddd-0403-4930-a5eb-](http://msdn.microsoft.com/ko-kr/library/7bd89ddd-0403-4930-a5eb-3c78718533d4)

[3c78718533d4](http://msdn.microsoft.com/ko-kr/library/7bd89ddd-0403-4930-a5eb-3c78718533d4))을 설정하여 보조 복제 노드에서 읽기작업을 수행하게 함으로써 주 노드의 과부하를 줄여주고 잠금 충돌도 줄일 수 있는 효과를 가져올 수 있다. 즉 서버의 개수만큼 충분히 작업에 활용 할 수 있는 것이다.

<p>주 노드 (DML)</p>		<pre>CREATE TABLE TEST ([t1 int identity, t2 char(1)) go INSERT INTO TEST(t2) VALUES('A') go 100</pre> <p>100 %</p> <p>메시지 실행 루프 시작</p> <p>(1개 행이 영향을 받음)</p> <p>(1개 행이 영향을 받음)</p>																								
<p>보조 노드 1 (SELECT)</p>		<pre>SELECT * FROM MG_130121.dbo.DB_FILE_INFO WHERE DB_NAME = 'maxgauge' UNION ALL SELECT * FROM MG_130122.dbo.DB_FILE_INFO WHERE DB_NAME = 'maxgauge' UNION ALL SELECT * FROM MG_130123.dbo.DB_FILE_INFO WHERE DB_NAME = 'maxgauge' go</pre> <p>100 %</p> <p>결과</p> <table border="1"> <thead> <tr> <th></th> <th>LOGTIME</th> <th>DB_NAME</th> <th>DB_SIZE</th> <th>DB_OWNER</th> <th>DB_ID</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2013-01-21 20:03:00,000</td> <td>maxgauge</td> <td>4,00 MB</td> <td>sa</td> <td>94</td> </tr> <tr> <td>2</td> <td>2013-01-22 21:14:12,000</td> <td>maxgauge</td> <td>4,00 MB</td> <td>sa</td> <td>94</td> </tr> <tr> <td>3</td> <td>2013-01-23 18:30:42,000</td> <td>maxgauge</td> <td>4,00 MB</td> <td>sa</td> <td>94</td> </tr> </tbody> </table>		LOGTIME	DB_NAME	DB_SIZE	DB_OWNER	DB_ID	1	2013-01-21 20:03:00,000	maxgauge	4,00 MB	sa	94	2	2013-01-22 21:14:12,000	maxgauge	4,00 MB	sa	94	3	2013-01-23 18:30:42,000	maxgauge	4,00 MB	sa	94
	LOGTIME	DB_NAME	DB_SIZE	DB_OWNER	DB_ID																					
1	2013-01-21 20:03:00,000	maxgauge	4,00 MB	sa	94																					
2	2013-01-22 21:14:12,000	maxgauge	4,00 MB	sa	94																					
3	2013-01-23 18:30:42,000	maxgauge	4,00 MB	sa	94																					
<p>보조 노드 2 (BACKUP)</p>		<p>이름 BACKUP</p> <p>설명 AlwaysON BACKUP</p> <table border="1"> <thead> <tr> <th>하위 계획</th> <th>설명</th> <th>일정</th> </tr> </thead> <tbody> <tr> <td>하위 계획_1</td> <td></td> <td>매주 월요일, 일요일 오...</td> </tr> </tbody> </table> <p>데이터베이스 백업(간체)</p> <p>로컬 서버 연결에서 데이터베이스 백업 데이터베이스: MG_130121.MG_130122.MG_130123 유형: 전체 기본 백업 추가 대상: 디스크 백업 압축 (Default)</p>	하위 계획	설명	일정	하위 계획_1		매주 월요일, 일요일 오...																		
하위 계획	설명	일정																								
하위 계획_1		매주 월요일, 일요일 오...																								

표6 AlwaysOn 복제 노드 활용방법 예

이제는 SQL Server 2012로 업그레이드가 필요한 때

앞에서 시나리오를 통해 간단히 살펴 본 것처럼 단순 Mirroring 과 비교했을 때도 AlwaysOn 은 많은 장점이 있다. 그 외의 HA 기능들과 비교했을 때도 AlwaysOn 은 충분히 우월하다고 본다. 물론 비용적인 측면에서 보면 이전버전을 쓰는 게 더 나을 수도 있다. 그렇지만 시대가 Big Data 나 Cloud Computing 이니 하며 시간이 지날수록 데이터의 중요성은 더욱 더 커지게 되고 지금 당장은 아니더라도 결국 (비용 < 데이터) 가 되는 시점이 발생할 것이다. 그렇다면 고민 없이 SQL server 2012 로 업그레이드 해서 데이터를 지키는 것이 정답이라고 생각한다. 그만큼 AlwaysOn 은 제 값을 하는 기능이라고 본다. 그 동안 HA 와 DR 을 위해 3rd Party Tool 을 이용해 비용이 이중으로 들었던 고객들에게 AlwaysOn 은 비용이나 기능적인 측면에서 희소식이 아닐까 생각해보면서 이 글을 마친다.