

Real Alternative DBMS ALTIBASE, Since 1999

ALTIBASE 백업정책 결정을 위한 고려사항

2014. 12



Copyright © 2000~2014 ALTIBASE Corporation. All Rights Reserved.

목차

개요	4
백업 방법의 종류	5
온라인 백업.....	5
오프라인 백업.....	5
백업 방법의 비교.....	6
백업 정책 결정을 위한 고려사항	7
공통적인 고려사항.....	7
iloader 백업 방식의 고려사항.....	8
dataport 백업 방식의 고려사항.....	9
full 백업 방식의 고려사항.....	10
증분 백업 방법의 고려사항.....	12
이중화를 이용한 백업 방법의 고려사항.....	15
오프라인 백업 방법의 고려사항.....	16
요약	17
안정성 측면에서의 비교.....	17
데이터 복구 가능 시점의 비교.....	17
백업에 필요한 저장 공간의 비교.....	18
사용자 입장의 관리 요소.....	18
복구 방식의 비교.....	19

개요

ALTIBASE가 제공하는 백업 방법들은 불시에 발생하는 장애에 대비해 별도의 물리적인 저장 공간에 복구 가능한 백업본을 만들어 내기 위해 사용된다. 복구는 백업본을 가지고 장애 이전 시점 또는 백업 시점까지의 DB상태로 만들어 내는 과정을 의미한다.

본 문서에서는 ALTIBASE가 제공하는 백업 방법들의 특징을 설명하고 어떤 사항들을 고려해 백업 정책을 운영할지 결정하는데 필요한 내용을 담고 있다.

본 문서는 ALTIBASE 버전 6.3.1 이상을 기준으로 작성되었다. 다음의 문서를 사전에 참고할 것을 권고한다.

1. 『ALTIBASE 용량 산정 가이드』
2. 『디스크 I/O를 고려한 볼륨 구성 가이드』
3. 『효율적인 이중화 구성 가이드』

백업 방법의 종류

ALTIBASE가 제공하는 백업 방법에 대해 설명한다. 본 문서에서는 DB객체의 생성 스크립트를 SQL문 형태로 저장하기 위해 사용되는 “aexport” 유틸리티는 포함하지 않는다. 물리적인 데이터 또는 데이터파일의 백업과 관련된 사항만을 설명한다.

ALTIBASE가 제공하는 백업 방법은 크게 프로세스가 운영 중인 상태에서 백업이 가능한지 여부에 따라 가능한 경우를 온라인 백업으로, 불가능한 경우를 오프라인 백업으로 나누어 설명한다.

온라인 백업

온라인 백업은 ALTIBASE 프로세스가 구동 중인 상태에서 DBA에 의해 또는 자동화된 스크립트와 같은 방법을 통해 현재의 DB를 백업하는 방식을 의미한다. 온라인 백업은 다시 운용 방법에 따라 크게 3 가지 형태로 나뉘며 각각은 백업 방법에 따른 세부 항목으로 나뉜다.

분류	백업 방법	설명
유틸리티 방식	iloader	ALTIBASE가 제공하는 테이블 단위의 백업 유틸리티인 iloader를 통해 개별 테이블의 스냅샷(snapshot)을 텍스트(text) 형태로 백업하는 것을 의미한다.
	dataport	테이블 단위의 백업을 위하여 ALTIBASE가 제공하는 시스템-정의 저장 프로시저인 dataport를 이용하여 테이블의 스냅샷(snapshot)을 바이너리(binary) 형태로 백업할 수 있다.
아카이브 방식	full 백업	DBA 또는 별도의 자동화된 스크립트 등을 통해 DB를 전체 또는 테이블스페이스 단위로 백업하는 방식을 기본으로 한다.
	증분 백업	처음 백업을 수행할 시에는 레벨 0 증분 백업을 반드시 수행해야 하며 이를 기준으로 이전 백업 이후에 변경된 페이지를 백업(레벨 1 증분 백업)하는 것을 기본으로 한다.
이중화 방식	이중화	ALTIBASE의 이중화 기능을 이용한 백업 방식으로 네트워크를 통해 트랜잭션 로그를 전송하여 상대방 DB에 데이터를 복제 시키는 방법으로, 하나의 이중화 객체에 다수의 테이블을 연동하여 사용한다.

오프라인 백업

오프라인 백업은 ALTIBASE 프로세스가 구동 중인 상태에서 수행할 수 없는 방법으로 ALTIBASE 프로세스의 구동 중지 상태에서 ALTIBASE 운영에 필요한 모든 파일을 사용자가 직접 복제 해야 한다. (이런 방법을 cold백업이라고 부르기도 한다.)

필요한 모든 파일에는 데이터파일, 로그엔커 파일, 트랜잭션 로그파일을 모두 포함한다. 사용자는 이러한 파일의 분산된 위치를 모두 파악해야 하며 운영체제가 제공하는 파일 복사 명령(ex. cp)을 이용하여 별도의 물리적인 디스크 공간에 복사함으로써 백업이 완료될 수 있다.

백업 방법의 비교

앞에서 언급한 백업 방법을 비교하여 설명한다.

분류	백업 방법	백업 수행 시점의 생성 파일	복구 가능 시점
온라인 백업	iloader	수행 시점에 사용자가 지정한 파일명으로 생성된 데이터파일	iloader를 실행한 시작 시점까지만 복구 가능
	dataport	수행 시점에 사용자가 지정한 파일명으로 생성된 데이터파일	dataport를 실행한 시작 시점까지만 복구 가능
	full 백업	데이터파일, 로그엔커 파일, 아카이브 로그파일	장애 발생 직전 시점까지 복구 가능
	증분 백업	데이터파일, 로그엔커 파일, 아카이브 로그파일, backupInfo 파일	장애 발생 직전 시점까지 복구 가능
	이중화	없음	상대편 서버가 가지고 있는 데이터로 복구 가능
오프라인 백업	물리적 복사	데이터파일, 로그엔커 파일, 트랜잭션 로그파일, altibase.properties 파일	복사를 수행한 시점까지만 복구 가능

여기서 full 백업과 증분 백업에서 “복구 가능 시점”을 “장애 발생 직전 시점까지”라고 설명 하는데 이는 현재의 트랜잭션 로그를 저장하고 있는 물리적인 디스크의 장애가 없을 경우에만 가능하다. 즉, 아카이브 방식으로 운영을 한다 하더라도 현재 쓰고 있는 트랜잭션 로그파일까지 아카이브 된 것을 보장하는 것이 아니기 때문에 트랜잭션 로그파일이 손상된 경우라면 장애 직전 시점까지 복구할 수 없는 경우도 존재할 수 있다.

증분 백업은 페이지 변경 추적 기능을 활성화 함으로써 사용할 수 있다. 이 기능을 활성화하면 변경된 페이지를 확인하는 changeTracking 파일과 증분 백업에 관한 정보가 기록되는 backupInfo 파일이 생성된다. 이 backupInfo 파일은 증분 백업 수행 시 데이터 파일과 함께 백업되며, backupInfo 파일이 소실될 경우 복구가 불가능하기 때문에 관리에 주의가 필요하다.

증분 백업의 아카이브 로그파일은 레벨 0 증분 백업 이후의 모든 아카이브 로그파일을 보관하고 있어야 특정 시점으로 복구가 가능하다.

이중화 방식은 백업 데이터 전송이 아닌 네트워크를 통한 HA(High Availability)구성을 말하는 것으로 동기화 방식에 따라 lazy와 eager 모드로 구분할 수 있다. lazy 모드는 지역 서버와 원격 서버의 트랜잭션이 별개로 동작하기 때문에 성능이 높지만, 경우에 따라 이중화 데이터 충돌에 의한 데이터의 불일치가 발생할 수 있다. eager 모드는 지역서버의 트랜잭션이 원격 서버의 트랜잭션에 반영을 확인한 후 동기화를 수행하기 때문에 성능은 매우 느리지만, 데이터의 일관성이 보장된다. ALTIBASE 이중화에 대한 자세한 내용은 『효율적인 이중화 구성 가이드』 문서를 참고한다.

백업 정책 결정을 위한 고려사항

백업 정책은 사용자가 운영 형태에 맞게 필요한 백업 방법과 백업 주기를 결정하는 것이다. 앞서 설명한 백업 방법들에 대해 좀 더 세세한 부분을 설명하고 각각 고려할 사항을 설명한다.

공통적인 고려사항

이하에 설명할 모든 백업 방법에서 공통적으로 고려할 사항을 설명한다.

고려사항	설명
백업을 위한 디스크	<p>full 백업과 증분 백업은 로컬 서버에 마운트(mount)된 디스크에만 기록할 수 있다.</p> <p>(full 백업의 BEGIN BACKUP ~ END BACKUP SQL 구문 사이에 넷 백업과 같은 백업 솔루션을 사용하여 테이프(tape)와 같은 저장 매체에 기록이 가능하다.)</p> <p>iloader 방식의 경우는 원격지에서 백업 대상 서버로 접속하여 수행할 수 있다.</p> <p>오프라인 백업의 경우는 원격지에서 ftp와 같은 방법을 통해 복제가 가능하다.</p>
백업 자동화	<p>ALTIBASE는 내부에 스케줄러 기능을 제공하며 사용자가 시스템의 crontab과 같은 자동화 툴을 이용하여 백업 스크립트를 작성하여 운용할 수 있으며, 지정된 시간에 자동으로 백업이 수행될 수 있도록 제어가 필요하다.</p> <p>넷 백업과 같은 3rd Party 백업 솔루션을 이용할 경우 스크립트를 통해 테이프(tape)와 같은 영역에 자동화된 백업 수행이 가능하다.</p>
디스크 I/O를 고려한 볼륨 구성	<p>백업을 수행하는 시점에 백업본 생성으로 다량의 디스크I/O가 발생할 수 있다. 이러한 백업 저장 공간을 일반적인 트랜잭션 공간과 동일하게 가져갈 경우 서비스의 응답 속도가 저하될 수 있으므로 백업의 저장 공간은 가능한 별도의 물리적으로 분리된 디스크 공간으로 설정하는 것을 고려해야 한다.</p> <p>이러한 공간의 고려가 안된 경우라면 서비스가 빈번하지 않은 시간을 선택하여 백업 작업이 수행될 수 있도록 고려해야 한다.</p>
데이터의 중요도를 고려한 백업 방식의 선택	<p>데이터의 중요도나 보관 주기에 따라 적절한 백업 방법을 선택해야 한다. 예를 들어, 당일 사용한 데이터를 매일 삭제하는 배치 작업이 돌아 가는 경우라면 해당 테이블까지 백업에 고려할 필요가 없는 경우도 존재하기 때문이다.</p> <p>또는, 중요 테이블만 일별 백업을 해도 가능한 경우이면서 동시에 테이블의 데이터가 크지 않은 경우라면 관리 포인트가 증가되는 아카이브 방식보다는 iloader백업 방식을 선택하는 경우도 가능하다.</p>
백업본의 보관 주기 및 저장 공간의	<p>정상적으로 만들어진 백업본이라 해도 사용자의 관리 실수로 해당 백업본으로 복구에 성공할 수 없는 예외적인 경우가 있을</p>

고려	<p>수 있다.(예를 들어 백업본 파일의 손상과 같은 이유.)</p> <p>따라서, 백업본은 마지막에 성공한 백업본과 함께 그 직전 백업본을 함께 저장/관리하는 것을 권장한다.</p> <p>최소 2벌의 백업본을 유지하도록 정책을 수립할 경우 마지막 백업본과 직전 백업본을 함께 저장 관리해야 할 것임으로 이에 대한 합산된 저장 공간을 필요로 한다. 이런 사항은 백업을 얼마나 자주 할 것인가에 대한 백업 주기 설정 및 로컬 서버 내의 보관 주기 및 방법에 따른 저장 공간을 고려해야 함을 의미한다.</p>
복구 시간 고려	<p>아카이브 방식은 백업 주기가 길어질 경우 복구에 필요한 시간이 늘어나게 된다. 복구 시 백업본과 백업 이후의 아카이브 로그파일을 이용하기 때문에 트랜잭션 양에 따라 복구 시간이 좌우된다.</p> <p>예를 들어 매월 초에 백업을 수행하고 월말에 장애가 발생할 경우 월초의 백업본과 월초 백업 이후 생성된 아카이브 로그파일을 모두 이용해야 하는데, 트랜잭션이 많은 경우, 적용해야 할 아카이브 로그파일이 많아지므로 복구하는데 오랜 시간이 소요될 수 있다.</p>

(중분 백업을 제외한 이하에서 설명하는 개별 백업 방법들에서는 백업본의 보관 주기와 관계 없이 한번의 백업 시점에 필요한 공간만을 설명한다.)

iloader 백업 방식의 고려사항

iloader의 경우는 테이블 단위로 백업을 수행하는 것이며 조건을 주어 필요한 영역의 데이터만 선별적으로 백업 하는 것도 가능하다. 따라서, 전체 백업이 필요하지 않고 특정 테이블이나 또는 테이블의 특정 영역만 백업을 해야 할 경우에 유용하다.(백업 시점에는 사용자가 지정한 텍스트 형태의 저장 파일이 생성된다.)

다만, 복구에 있어서는 ALTIBASE가 정상 구동 되어 있어야 하며 백업 시점에 만들어진 저장 파일을 가지고 테이블 단위로 복구해야 하기 때문에 백업 시점까지만 복구가 가능하다. 따라서, 한번 백업을 수행한 이후 다음 백업 시점까지 변경된 데이터는 복구를 할 수 없다는 단점을 가지고 있다.

고려사항	설명
GC의 증가	<p>이 사항은 ALTIBASE의 동시성 제어 기법인 MVCC(Multi Versioning Concurrent Control)와 관련이 있다. (MVCC와 관련된 자세한 기술적인 사항은 『ALTIBASE MVCC & GC』 문서를 참고하도록 한다.)</p> <p>백업 시점에 트랜잭션이 빈번할 경우 MVCC기법에 의해 불필요하게 생성된 레코드가 증가할 수 있으며 이는 메모리 증가 현상을 유발할 가능성이 있다.</p>
디스크 용량	<p>iloader에 생성되는 텍스트 저장 파일은 데이터뿐만 아니라 구분자를 필요로 한다. 구분자는 칼럼 구분자와 레코드 구분자로 구성된다.</p> <p>백업 공간의 예상 크기 (테이블 별 사용량 + 구분자의 용량) = SUM (</p>

	<p>(테이블 용량 * 보정치 1.2[수치 데이터 칼럼에 대한 보정치]) + (레코드 개수 * 레코드 구분자의 길이) + (레코드 개수 * (칼럼 개수-1) * 칼럼 구분자의 길이))</p> <p>* 수치 데이터 칼럼에 대한 보정치는 integer와 같은 4byte 형태의 칼럼이라도 텍스트로 저장되는 시점에는 4byte 이상일 수 있으므로 이에 대한 보정치를 적용하는 것을 의미한다.</p>
<p>성능을 위한 옵션</p>	<p>iloader를 이용한 백업 복구 시 아래와 같은 옵션을 이용하면 좀 더 성능이 향상될 수 있다.</p> <p>- array: 데이터 업로드(in) 시 파일에서 읽은 데이터를 배열로 구성하여 서버로 전송한다. 이는 통신 비용을 줄여 성능을 향상시킬 수 있다.</p> <p>- commit: 데이터 업로드(in) 시 몇 건 단위로 커밋 할 것인지 나타낸다. -array 옵션과 함께 사용할 시 array_size*commit 만큼 삽입 후 커밋한다.</p> <p>- atomic: 데이터 업로드(in) 시 Atomic Array Insert를 수행한다. 배열 크기만큼 Insert문을 하나의 트랜잭션으로 처리하여 성능이 빠르다. 이 옵션은 -array 옵션과 함께 사용해야 한다.</p> <p>- parallel: 병렬로 처리할 스레드의 개수를 지정한다. 다운로드(out) 시 설정한 값만큼 파일이 생성되며 -array 옵션과 함께 사용해야 한다. 업로드(in) 시 설정 값+1 만큼 연결을 생성한다.</p> <p>- split: 데이터 다운로드(out) 시 파일마다 저장할 레코드 개수를 설정한다. 파일이 나누어 생성되므로 업로드(in) 시 병렬처리가 가능하다.</p> <p>예를 들어 아래와 같이 사용할 수 있다.</p> <p>데이터 다운로드 : <code>iloader -s localhost -u SYS -p MANAGER out -f SYS_T1.fmt -d SYS_T1.dat -split 1000000 -log SYS_T1.log</code></p> <p>데이터 업로드 : <code>iloader -s localhost -u SYS -p MANAGER in -f SYS_T1.fmt -d SYS_T1.dat -t "^G_G^" -r "^R_R^n" -array 1000 -atomic -commit 100 -parallel 4 -log SYS_T1.log -bad SYS_T1.bad</code></p> <p>ALTIBASE iloader에 대한 자세한 내용은 『iLoader』 메뉴얼을 참고한다.</p>

dataport 백업 방식의 고려사항

dataport의 경우 테이블 단위로 백업을 수행하며 시스템-정의 저장 프로시저 형태로 제공된다. 따라서, 전체 백업이 필요하지 않고 특정 테이블의 데이터만 백업 받을 경우 유용하다. dataport는 QP(Query Process)를 거치지 않고 바이너리 형태로 데이터를 마이그레이션하기 때문에 iloader 보다 성능이 빠르지만 몇 가지 제약 사항이 있으므로 반드시 확인 후 백업을 결정해야 한다. (백업 시점에는 사용자가 지정한 바이너리 형태의 저장 파일이 생성된다.)

dataport로 백업한 데이터 파일에 dumpdp 유틸리티를 적용하면 테이블 생성 구문과 데이터 입력 구문을 출력해주므로 스크립트를 따로 관리하여 필요 시에 해당 데이터만

확인하여 쿼리를 수행할 수 있다. (인덱스와 그 외 정보들은 백업되지 않으므로 주의한다.)

다만, 복구에 있어서는 ALTIBASE가 정상 구동 되어 있어야 하며 백업 시점에 만들어진 저장 파일을 가지고 테이블 단위로 복구해야 하기 때문에 백업 시점까지만 복구가 가능하다. 따라서, 한번 백업을 수행한 이후 다음 백업 시점까지 변경된 데이터는 복구를 할 수 없다는 단점을 가지고 있다.

고려사항	설명
GC의 증가	이 사항은 ALTIBASE의 동시성 제어 기법인 MVCC(Multi Versioning Concurrent Control)와 관련이 있다. (MVCC와 관련된 자세한 기술적인 사항은 『ALTIBASE MVCC & GC』 문서를 참고하도록 한다.) 백업 시점에 트랜잭션이 빈번할 경우 MVCC기법에 의해 불필요하게 생성된 레코드가 증가할 수 있으며 이는 메모리 증가 현상을 유발할 가능성이 있다.
디스크 용량	dataport를 이용한 테이블의 데이터는 바이너리 파일로 생성되는데 데이터에 어떤 변환도 하지 않고 그대로 export하고 import한다. 백업 데이터 파일이 저장되는 디스크가 부족할 경우 백업이 정상적으로 이루어질 수 없으므로 여유있는 공간을 활용하는 것이 좋다.
제약 사항	dataport를 사용할 경우 데이터 업로드 시 아래와 같은 제약사항이 존재하므로 확인 후 대안을 설정하여 백업을 결정해야 한다. 1. 업로드 할 테이블이 존재해야 한다. 2. 원본 테이블과 대상 테이블의 모든 속성이 동일해야 한다. 3. 암호화된 컬럼을 포함하는 테이블은 사용할 수 없다. 4. 대상 테이블에 트리거나 인덱스가 있거나 제약조건(not null, foreign key)을 가지고 있는 경우 사용할 수 없으므로 제거 후 사용해야 한다. 5. 이중화 대상 테이블일 경우 사용할 수 없다. * dataport를 이용한 데이터 다운로드 시 인덱스, 제약조건, 트리거는 백업되지 않는다.

full 백업 방식의 고려사항

full 백업 방식은 DB의 설정 값을 아카이브 로그모드로 변경함으로써 가능하다. 설정값의 변경은 ALTIBASE를 구동하는 단계에서만 변경이 가능하며 변경한 이후부터는 트랜잭션 로그파일이 스위칭(switching = 현재 기록 중인 트랜잭션 로그파일이 모두 기록된 후 다음 트랜잭션 로그파일이 기록되는 시점) 될 때마다 가장 오래된 트랜잭션 로그파일부터 자동으로 사용자가 지정한 아카이브 저장 디렉토리로 복제가 된다.

full 백업 방식은 사용자가 수행한 백업 SQL문을 통해 데이터파일과 로그엔커 파일이 사용자가 지정한 백업 디렉토리로 저장된다.

full 백업 방식의 복구 시점에는 저장된 (필요 시)아카이브 로그파일과 백업 SQL문에 의해 복제된 데이터파일, (필요 시)로그엔커 파일 등을 이용하여 복구가 가능하다.

백업되는 파일을 다시 정리하면 아래와 같다.

백업 파일	설명
데이터파일	사용자가 수행한 백업 SQL문을 통해 지정된 디렉토리로 복제됨
로그앵커 파일	
아카이브 로그파일	자동으로 ALTIBASE 내부의 쓰레드에 의해 복제된다.

full 백업 방식에서는 테이블스페이스 별로 백업을 수행할 수 있으며 BEGIN BACKUP ~ END BACKUP SQL 구문을 수행하는 사이에 데이터파일에 대한 복제를 허용하는 방법도 제공된다. (백업 받는 테이블스페이스에 대한 체크포인트를 막아 백업 중 데이터의 변경을 제한한다.)

full 백업 방식에서는 다음의 사항을 고려해야 한다.

고려사항	설명
아카이브 로그파일에 대한 저장 공간	<p>아카이브 로그파일은 사용자가 삭제하기 전까지는 유지된다. 따라서, 관리하지 않은 경우 디스크 용량이 부족한 문제가 발생할 수 있는 점을 고려해야 한다.</p> <p>아카이브 로그파일의 삭제 가능 여부는 백업이 정확히 완료된 시점에 백업된 로그앵커의 "delete logfile(s) range" 정보를 통해 판단할 수 있으나 일반적으로 사용자가 판단하기는 어렵다. 따라서, 백업이 완료된 시점에 아카이브 로그파일은 별도의 테이프와 같은 공간에 복제한 후 삭제하는 형태를 취한다.</p> <p>임의로 삭제한 상태에서 만일 최종 백업본이 복구 상황에서 문제가 될 경우 그 직전에 백업된 파일을 가지고 복구해야 한다면 직전에 삭제했던 아카이브 로그파일이 다시 필요할 수 있기 때문이다.</p>
백업본의 저장 공간	<p>백업을 수행하여 생성된 데이터파일은 메모리DB의 경우 평평 체크포인트 방식으로 인해 두 별의 데이터파일이 저장되는데 백업 시점에는 한 별의 데이터파일만 저장된다. 따라서, 메모리DB는 한 별의 데이터파일의 저장 공간을 고려하면 된다.</p> <p>디스크DB의 경우는 현재 사용 중인 모든 데이터파일이 저장될 공간을 산정해야 한다. (임시성 테이블스페이스에 속한 데이터파일은 배제)</p> <p>따라서, Full 백업 방식에서의 백업본의 저장 공간은 용량 산정에서 예측된 용량에서 다음과 같이 계산할 수 있다.</p> <p>백업본 저장 공간 = (메모리 DB 1 별의 예측 공간) + (디스크 DB의 예측 공간) + (언두 테이블스페이스의 예측 공간)</p>
트랜잭션 로그파일의 저장 공간	<p>백업이 수행되는 동안에는 체크포인트가 동시에 수행될 수 없다. 따라서, 백업이 완료될 때까지는 변경 트랜잭션에 의해 트랜잭션 로그가 지속적으로 쌓이게 된다.</p> <p>트랜잭션 로그파일은 체크포인트가 완료된 시점에만 삭제가 가능하기 때문에 Full 백업 방식을 고려할 경우에는 백업이 완료되는 시간을 고려하여 트랜잭션</p>

	로그파일의 디스크 저장 공간도 여유 있게 설정해야 할 필요가 있다.
--	---------------------------------------

백업과 관련된 용량의 산정 부분은 『ALTIBASE 용량 산정 가이드』를 참고하도록 한다.

중분 백업 방법의 고려사항

중분 백업 방식은 아카이브 로그모드에서만 사용이 가능하며 변경된 페이지를 추적하기 위해 페이지 변경 추적 기능을 활성화해야 한다. 중분 백업 방식에서 페이지 변경 추적 기능을 수행하면 레벨 0 중분 백업이 수행될 때 비로소 활성화된다.

페이지 변경 추적 기능이 활성화 될 때 changeTracking 파일과 backupInfo 파일이 생성되고 비활성화 시 changeTracking 파일은 삭제되고 중분 백업은 사용할 수 없게 된다. 다시 페이지 변경 추적 기능을 활성화하면 중분 백업은 사용할 수 있지만 기존에 받은 백업 본은 사용할 수 없으므로 다시 백업을 수행해야 한다. 또한 backupInfo 파일은 소실되면 복구가 불가능하기 때문에 철저히 관리해야 한다.

중분 백업을 위한 페이지 변경 추적 기능을 활성화하면 변경 페이지를 추적하는 부하가 발생되므로 DB의 성능 저하가 발생하는 점을 반드시 염두에 두어야 한다.

중분 백업은 아래와 같이 2 개의 레벨로 구성된다.

분류	설명
레벨 0 중분 백업	full 백업의 전체 백업과 같이 모든 데이터파일을 백업 받는다. 레벨 0 중분 백업은 중분 백업을 사용하기 위해 반드시 한번 이상 수행해야 하며 중분 백업의 기준이 된다. 효율적인 백업을 위해서는 레벨 0 중분 백업을 주기적으로 수행해야 한다. (full 백업과 데이터 양은 같지만 용도가 full 백업/ 중분 백업으로 구분되어 있으므로 서로 대체하여 사용할 수 없다.)
레벨 1 중분 백업	이전 중분 백업 이후 변경된 페이지만 백업한다. 변경된 페이지만 백업하므로 백업 파일의 크기가 작고 백업 시간이 적게 걸린다.

특히 레벨 1 중분 백업은 아래와 같이 2 가지 형태로 나뉜다.

분류	설명
차등 중분 백업 (default)	가장 최근에 수행된 레벨 0 또는 레벨 1 중분 백업 이후 변경된 페이지를 백업한다. 차등 중분 백업은 누적 중분 백업에 비해 백업의 양은 작지만 복구 시 여러 개의 백업 파일을 이용하므로 복구 시 시간이 더 소요된다.
누적 중분 백업	가장 최근에 수행된 레벨 0 중분 백업 이후의 변경된 페이지를 백업한다. 누적 중분 백업은 차등 중분 백업에 비해 백업 크기가 크고 시간이 더 걸리지만 복구 시간을 단축할 수 있다.

중분 백업을 사용하기 위해서는 아래와 같이 레벨 0 중분 백업과 차등 중분 백업, 누적 중분 백업을 적절히 혼용하여 백업 계획을 세워야 한다.

	일	월	화	수	목	금	토	일
레벨 0 증분 백업	←							
레벨 1 차등 증분 백업		←						
레벨 1 차등 증분 백업			←					
레벨 1 누적 증분 백업		←	←	←	←	←	←	←
레벨 1 차등 증분 백업					←			
레벨 1 차등 증분 백업						←		
레벨 1 누적 증분 백업		←	←	←	←	←	←	←
레벨 0 백업	←							

레벨 0 증분 백업을 너무 자주 수행하면 full 백업과 백업 내용이 다르지 않기 때문에 비효율적이며 레벨 0 증분 백업 주기가 너무 길면 관리해야 할 레벨 1 증분 백업본과 아카이브 로그파일의 개수가 증가하므로 복구에 소요되는 시간도 늘어난다. 따라서 증분 백업을 수행할 경우에는 반드시 업무와 데이터 양에 따라 적절한 백업 계획이 필요하다.

증분 백업의 복구 시점에는 저장된 (필요 시)아카이브 로그파일과 백업 SQL문에 의해 복제된 데이터파일, backupInfo 파일, (필요 시)로그엔커 파일 등을 이용하여 복구가 가능하다.

백업되는 파일을 다시 정리하면 아래와 같다.

백업 파일	설명
데이터파일	사용자가 수행한 백업 SQL문을 통해 지정된 디렉토리로 복제됨
로그엔커 파일	
백업 정보 파일	
아카이브 로그파일	자동으로 ALTIBASE 내부의 쓰레드에 의해 복제된다.

증분 백업은 full 백업과 마찬가지로 테이블스페이스 별로 백업을 수행하는 SQL문이 지원되지만 BEGIN BACKUP ~ END BACKUP SQL 구문이 지원되지 않는다.

증분 백업 방식에서는 다음의 사항을 고려해야 한다.

고려사항	설명
아카이브 로그파일에 대한 저장 공간	<p>아카이브 로그파일은 사용자가 삭제하기 전까지는 유지된다. 따라서, 관리하지 않은 경우 디스크 용량이 부족한 문제가 발생할 수 있는 점을 고려해야 한다.</p> <p>아카이브 로그파일의 삭제 가능 여부는 백업이 정확히 완료된 시점에 백업된 로그엔커의 “delete logfile(s) range” 정보를 통해 판단할 수 있으나 일반적으로 사용자가 판단하기는 어렵다. 따라서, 백업이 완료된 시점에 아카이브 로그파일은 별도의 테이프와 같은 공간에 복제한 후 삭제하는 형태를 취한다.</p> <p>증분 백업의 경우 특정 시점으로 복구하려 할 때 특정</p>

	<p>시점 이전의 백업본과 특정 시점 사이의 아카이브 로그가 필요하게 되는데 이러한 불완전 복구를 위해서는 마지막 증분 백업 이후의 아카이브 로그파일을 모두 보관해야 한다. 이는 full 백업 방식보다 아카이브 로그파일의 보관주기가 길어질 수 있고 저장 공간이 늘어날 가능성이 있으므로 충분히 고려해야 한다.</p>
<p>백업본의 저장 공간</p>	<p>증분 백업을 수행하여 생성된 데이터파일은 메모리DB의 경우 백업 시점에 한 벌의 데이터파일만 저장이 된다. 따라서 메모리DB는 한 벌의 데이터파일의 저장 공간을 고려하면 된다.</p> <p>증분 백업의 디스크DB는 현재 사용 중인 모든 데이터파일이 저장될 공간을 산정해야 한다. (임시성 테이블스페이스에 속한 데이터파일은 배제)</p> <p>따라서, 증분 백업에서 백업본의 저장 공간은 용량 산정에서 예측된 용량에서 다음과 같이 계산할 수 있다.</p> <p>백업본 저장 공간 = (메모리 DB 1 벌의 예측 공간) + (디스크 DB의 예측 공간) + (언두 테이블스페이스의 예측 공간)</p> <p>레벨 1 증분 백업은 changeTracking 파일에 기록된 변경 페이지를 백업하기 때문에 업무에 따른 사용량이나 패턴에 따라 백업의 크기가 천차만별이다.</p> <p>따라서 반드시 증분 백업에 대한 테스트를 통해 용량을 산정한 후 증분 백업을 수행해야 한다.</p>
<p>트랜잭션 로그파일의 저장 공간</p>	<p>백업이 수행되는 동안에는 체크포인트가 동시에 수행될 수 없다. 따라서, 백업이 완료될 때까지는 변경 트랜잭션에 의해 트랜잭션 로그가 지속적으로 쌓이게 된다.</p> <p>트랜잭션 로그파일은 체크포인트가 완료된 시점에만 삭제가 가능하기 때문에 증분 백업 방식의 백업을 고려할 경우에는 백업이 완료되는 시간을 고려하여 트랜잭션 로그파일의 디스크 저장 공간도 여유 있게 설정해야 할 필요가 있다.</p>
<p>백업 파일 관리</p>	<p>증분 백업을 위해 생성되는 changeTracking 파일은 메모리 이미지파일과 동일한 경로에 생성되는데, 해당 경로의 공간이 부족하게 되면 정상적인 페이지 변경 추적 기능을 사용할 수 없으며, 이는 백업이 정상적으로 동작할 수 없다는 것을 뜻한다. 또한 changeTracking 파일이 삭제되면 증분 백업을 사용할 수 없으며 페이지 변경 추적 기능을 활성화하여 changeTracking 파일을 다시 생성한 후 레벨 0 증분 백업부터 백업을 진행해야 한다.</p> <p>페이지 변경 추적 기능은 증분 백업을 사용하기 위해 반드시 필요하지만 사용하면 DB에 성능 저하를 가져온다는 점을 반드시 염두에 두어야 한다.</p> <p>backupInfo 파일은 메모리 데이터파일과 함께 백업되며,</p>

	복구시에도 사용되기 때문에 백업본에 backupInfo 파일이 없을 경우 복구가 불가능하다.
--	---

백업과 관련된 용량의 산정 부분은 『ALTIBASE 용량 산정 가이드』를 참고하도록 한다.

이중화를 이용한 백업 방법의 고려사항

이중화를 이용한 백업의 개념은 로컬 서버에 위치한 별도의 디스크 공간에 파일 형태로 백업하는 개념이 아닌 원격 서버로 데이터를 전송/복제하여 백업하는 개념으로 이해를 하면 된다. 일반적인 시스템 구성은 단독 서버 구성이 아닌 장애를 대비한 HA(High Availability)구성을 많이 하게 되는데 이러한 구성으로 백업의 효과를 동시에 달성하는 형태이다.

하지만, ALTIBASE가 제공하는 lazy 모드의 이중화 방식은 다양한 요인에 의해 네트워크 지연/불능이 발생할 수 있고 장애 직전에 미처 보내지 못한 데이터가 존재할 수 있다. 이로 인해 로컬 서버의 데이터와 원격 서버에서 서비스 중에 변경되는 데이터가 서로 다른 값으로 존재하는 경우가 발생할 수 있다. 따라서, 이러한 상황을 고려해도 되는 서비스라면 별도의 디스크 공간을 필요로 하는 백업이 아닌 이중화를 통한 백업 구축을 고려할 수 있다.

이중화는 데이터 복구 수행 시 SYNC 명령어를 통해 이중화 대상 테이블에 대한 복구를 수행할 수 있으며 필요한 테이블 단위로 SYNC 기능을 사용할 수 있다.

추가적인 고려사항은 2대의 서버에 동시에 디스크 장애가 발생하여 복구 불능 상태가 발생하는 가능성도 존재할 수 있음을 고려해야 한다.

고려사항	설명
서비스 고려	백업이 필요한 테이블은 각각의 서버에 해당 테이블이 존재해야 하고 이중화로 구성이 되어야 한다.
이중화 모드	이중화는 전송 방식에 따라 lazy 모드와 eager 모드로 구분한다. lazy 모드는 데이터의 불일치 가능성이 있으나 성능이 빠르다. 이중화 지연 및 데이터의 충돌로 인한 데이터 불일치가 발생하지 않도록 서비스 구성을 고려해야 한다. eager 모드는 데이터의 불일치는 발생하지 않으나 성능이 크게 저하된다는 점을 유의해야 한다.
계약 사항	이중화를 생성하기 위한 제약 사항을 미리 고려한 후 서비스를 구성한다. 1. 이중화 대상 테이블은 기본키(primary key)가 반드시 생성 되어야 한다. 2. 기본키(primary key)는 업데이트를 허용하지 않는다. 3. 이중화 대상 테이블은 외래키(foreign key)를 가질 수 없다. 4. 이중화 대상 테이블은 DDL 작업이 불가능하다. 5. 시퀀스(sequence)는 이중화 되지 못한다. 6. 이중화 대상 테이블에 트리거(trigger)를 생성할 수 없다.

ALTIBASE 이중화와 관련해서는 『효율적인 이중화 구성 가이드』 문서를 참고한다.

오프라인 백업 방법의 고려사항

오프라인 백업은 ALTIBASE를 운영하는데 필요한 모든 파일을 별도의 공간에 복사하는 방식이다. 운영체제의 특성상 프로세스가 살아 있는 동안에 파일을 복사할 경우 복사된 파일의 내용을 보장할 수 없기 때문에 반드시 ALTIBASE를 중지 시킨 이후 파일을 복사하도록 해야 한다.

이 방식은 ALTIBASE를 중단해야 하기 때문에 필연적으로 서비스의 중단을 필요로 한다. 사용자는 파일을 복사할 시간만큼의 서비스 중단이 가능한 경우라면 백업의 한 방법으로 고려할 수 있다.

iloader와 마찬가지로 백업 시점까지만 데이터의 복구가 가능하며 서비스를 중단한 경우임으로 백업의 빠른 완료를 위한 디스크I/O의 성능 및 용량 부분만 고려하면 된다.

고려사항	설명	
디스크 용량	백업이 필요한 파일의 목록	
	데이터파일	v\$DATAFILES에 조회되는 모든 파일 v\$MEM_TABLESPACE_CHECKPOINT_PATHS에 조회되는 디렉토리에 위치한 모든 파일
	로그앵커 파일	\$ALTIBASE_HOME/conf/altibase.properties 파일 내에 "LOGANCHOR_DIR" 에 정의된 디렉토리에 존재하는 모든 파일
	트랜잭션 로그파일	\$ALTIBASE_HOME/conf/altibase.properties 파일 내에 "LOG_DIR" 에 정의된 디렉토리에 존재하는 모든 파일
	프로퍼티 파일	\$ALTIBASE_HOME/conf/altibase.properties 파일
	백업 공간의 예상 크기 =SUM (위의 파일 목록에 취합된 모든 파일의 크기)	

오프라인 백업에서는 백업해야 하는 트랜잭션 로그파일의 개수를 줄이기 위해 ALTIBASE 구동 정지 전 반드시 체크포인트를 명시적으로 수행하도록 한다. 그렇게 함으로서 복사되는 파일의 개수를 줄이고 향후에 복구에 필요한 시간도 절약할 수 있다.

오프라인 백업을 수행할 때 ALTIBASE의 altibase.properties 파일을 함께 백업하여 설정파일이 소실될 경우에 기존의 프로퍼티 설정이 적용될 수 있도록 대비할 필요가 있다.

요약

앞에서 설명한 대로 사용자는 디스크의 공간, 데이터의 중요도, 백업본의 보관 주기 및 백업 주기와 같은 요소들을 다양하게 고려하여 백업 정책을 결정해야 한다. 중요한 것은 현재의 구성된 시스템에서 충분한 백업 공간이 확보 가능한지 여부와 선택한 백업 방법으로 장애 이후 원하는 수준까지의 복구가 가능한지 여부를 판단하는 것이다.

안정성 측면에서의 비교

복구를 위해서는 반드시 데이터 파일 뿐 아니라 트랜잭션 로그파일과 로그엔커 파일을 필요로 한다. 이런 관점에서 모든 필요 파일들에 대한 최대한의 실시간 백업이 가능한지와 복구 가능 여부가 안정성을 판단하는 기준이 된다.

그런 관점에서는 Full 백업 방식이 가장 높은 안정성을 가지고 있다. 여타의 백업 방식은 현재의 트랜잭션 로그파일이 유실 되는 경우를 대처할 대안이 없기 때문이다. 만일, Full 백업 방식 이외의 백업 정책을 결정한 상황에서 트랜잭션 로그파일이 손상된 경우에는 최악의 경우 DB를 새롭게 생성한 후 데이터를 복구해야 한다.

데이터 복구 가능 시점의 비교

복구 가능 시점의 관점에서 각각의 방법으로 복구 가능한 시점은 아래와 같이 나눌 수 있다. (백업 시점을 검은색 점선으로 장애 직전 시점을 빨간색 점선으로 구분한다.)

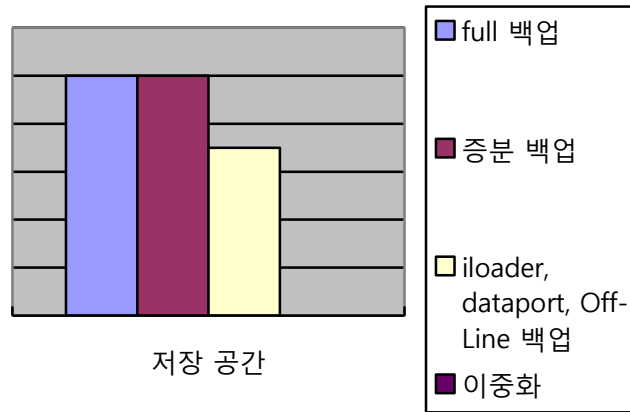
분류	백업 시점	장애 시점
Full 백업, 증분 백업 방식	→	→
iloader, dataport, 오프라인 백업 방식	→	→
이중화 방식	→	→

iloader와 dataport, 오프라인 백업 방식은 백업을 수행한 시점까지만 복구가 가능하지만 만일, 사용자가 그러한 문제를 회피할 수 있는 경우가 가능하다면 선택할 수 있다.

일례로 증권사의 시세 서버에 적용된 ALTIBASE의 경우는 모두 iloader를 통한 특정 history성 테이블만을 백업한다. 서버의 구성 상 장애가 발생한 시점에 다른 서버가 즉각 서비스 할 수 있으며 야간에 서비스를 중지하고 주요 테이블은 매일 배치를 통해 삭제되고 history성 테이블만 복구하면 되기 때문이다. 이러한 history성 테이블은 일별로 각각의 서버가 백업을 수행하고 장애 시 복구 후 최종 증가 정보만 입력되면 되기 때문에 iloader를 통한 백업 및 복구가 서비스에 전혀 지장이 없는 경우라고 볼 수 있다.

백업에 필요한 저장 공간의 비교

백업본의 저장 공간 예측 크기에 따라 아래와 같이 나눌 수 있다.



Full 백업 방식과 증분 백업 방식은 아카이브 로그파일의 저장 공간을 필요로 하기 때문에 여타 백업 방식에 비해서는 더 많은 저장 공간을 필요로 한다.

증분 백업 방식은 단기적으로 보았을 때는 Full 백업과 크게 차이가 없지만 장기적으로 보면 Full 백업 방식보다 데이터파일에 대한 저장 공간을 적게 필요로 할 수 있다. 하지만 레벨 0 증분 백업 이후 아카이브 로그파일을 모두 보관해야 하므로 백업 보관 주기와 데이터의 크기에 따라 백업의 양이 매우 유동적이므로 용량 산정 및 백업 계획을 위해서 사전 테스트가 반드시 필요하다.

iloader 방식과 dataport 방식은 전체 테이블의 백업 여부에 따라 유동적이므로 저장 공간의 차이가 크다. 백업 공간에 대한 예측은 백업 방식이나 상황에 따라 차이를 보일 수 있으므로 일반적인 비교 정도로 참고하도록 한다.

(위의 표는 모든 백업 방식에 대해 백업을 한번 수행하였을 때를 기준으로 한다.)

사용자 입장의 관리 요소

사용자는 백업 정책과 무관하게 실 운영 시스템의 디스크의 가용 공간을 항상 모니터링 해야 하는데 여기에 백업으로 추가적인 관리 요소가 생기게 된다.

분류	관리 요소
iloader	백업본의 저장 가용 공간
dataport	백업본의 저장 가용 공간
Full 백업	아카이브 로그파일의 저장 가용 공간, 백업본의 저장 가용 공간
증분 백업	아카이브 로그파일의 저장 가용 공간, 백업본의 저장 가용 공간
이중화	이중화의 상태(트랜잭션 로그파일의 공간)
오프라인 백업	백업본의 저장 가용 공간

복구 방식의 비교

각각의 백업을 가지고 간략하게 복구하는 방법을 비교, 설명한다.

분류	복구
iloader	텍스트 파일 형태로 받은 데이터를 iloader 구문을 이용하여 복구한다. 테이블은 미리 생성되어야 하며 테이블에 대한 데이터만 복구할 수 있다.
dataport	iSQL로 접속하여 프로시저 수행을 통해 데이터를 업로드 한다. 테이블은 미리 생성되어야 하며 테이블에 대한 데이터만 복구할 수 있다. 추가적으로 dumpdp 를 이용하여 테이블 생성과 데이터 입력을 위한 쿼리문의 추출이 가능하며, 테이블 생성 구문에는 인덱스와 제약조건 등의 추가사항은 포함되지 않는다.
Full 백업	복구할 시점이나 방법에 따라 백업한 데이터파일과 아카이브 로그파일, 트랜잭션 로그파일이 필요하다. 복구 시에는 control 단계에서 recover 명령어를 사용하여 복구한다.
중분 백업	복구할 시점이나 방법에 따라 백업한 데이터파일과 아카이브 로그파일, 트랜잭션 로그파일, backupInfo 파일이 필요하다. 복구 시에는 control 단계에서 restore, recover 명령어를 사용하여 복구한다.
이중화	이중화는 SYNC 명령어를 이용하여 이중화 테이블에 대한 데이터를 복구할 수 있다. (두 대의 장비 중 한 장비의 데이터를 기준으로 복구하는 방법이다.)

ALTIBASE[®]

알티베이스주

서울특별시 구로구 구로 3 동 182-13
대릉포스트 2 차 1008 호
02-2082-1000
<http://www.altibase.com>

기술본부

서울특별시 구로구 구로 3 동 182-13
대릉포스트 2 차 908 호
02-2082-1114

기술지원센터

02-2082-1114
<http://support.altibase.com>

Copyright © 2000~2014 ALTIBASE Corporation. All Rights Reserved.

이 문서는 정보 제공을 목적으로 제공되며, 사전에 예고 없이 변경될 수 있습니다. 이 문서는 오류가 있을 수 있으며, 상업적 또는 특정 목적에 부합하는 명시적, 묵시적인 책임이 일체 없습니다. 이 문서에 포함된 ALTIBASE 제품의 특징이나 기능의 개발, 발표 등의 시기는 ALTIBASE 재량입니다. ALTIBASE는 이 문서에 대하여 관련된 특허권, 상표권, 저작권 또는 기타 지적 재산권을 보유할 수 있습니다.