시스템 관리자와 함께 하는 테마여행 3

이진철 jincheol.lee@goodus.com

현재 굿어스(주) TM사업부에서 System Administrator와 DBA 업무를 담당하고 있으며, 주로 대용량 데이터베이스 시스템의 튜닝 및 컨설팅을 하고 있다. 굿어스(주)는 IT Infrastructure 전반에 대하여 OutSourcing 과 OutTasking을 제공하는 전문 IT Service 회사로써, Network, Server, Database 분야에 대한 Maintenance, Operation, Management를 제공한다.

인터넷 의존도가 높아감에 따라 'Web time is Real Time'이라는 말은 이미 우리 생활의 필수 요구사항이 되었다. 갈수록 사용자들은 참을성을 잃어가고, 다양한 네트워크와 데이터 서비스 환경 하에서 장애(Crash) 처리에 소요되는 시간(MTTR :Mean Time to Recover)은 이제 촌각을 다투고 있다. 어쩌면 시스템 관리자에게 가장 귀찮은 일 중 하나인 백업과 복구에 대한 기술은 매우 복잡하게 얽혀 있으며, 반드시 하지 않으면 안되는 업무분야이기도 하다.

이 글은 3회에 걸쳐 연재되고 있는 **System Administration** 분야 중 5장과 6장에 해당되는 내용을 다루고 있다.

1. 시스템 유지보수(System Maintenance) 일반론
2. 성능관리(Performance Management) 분야
3. 가용성 관리(Availability Management) 분야
4. 용량 관리(Capacity planning & Sizing) 분야
5. 장애 복구(Crash Recovery) 분야.
6. 서비스 엔지니어의 관점에서 바라보는 ITIL의 적용

11월호에서 다루었던 성능관리(Performance Management) 분야는 어찌보면 시스템 엔지니어들에게는 흥미롭고 재미있는 분야이기도 하다. 성능 상 문제가 있는 시스템을 튜닝하여 고객이 원하는 Response Time을 이끌어 내는 업무는 속된 말로 "잘하면 칭찬(?), 못해도 본전(?)"이기 때문이다. 그러나, 시스템의 백업과 복구 처리에서 "본전(?)" 이라는 말은 용납되지 않는다. 장애가 발생하여 Service가 중단된 것도 회사에 커다란 손실로 작용하는데, 복구마저 할 수 없는 상황이라면, 이러한 사태를 초래한 시스템 엔지니어는 도태될 것이 분명하다.

이번 호에서 다루는 장애 복구 처리는 가용성의 연장선상에서 Resiliency를 고려한 분야들을 다루고자 한다. "고가용성의 연장선상에서의 복제기술", 그리고, 재해 복구 시스템"에 대해서 전반적인 분야들을 다룰 것이며 근본적인 원칙들을 생각해 볼 것이다.

마지막으로, System Engineer의 Service Process를 관리하는 한 방법으로 ITIL 이라는 Best Practice의 적용에 관하여 개괄적으로 논하고자 한다.

5. 장애 복구(Crash Recovery) 분야

고가용성의 연장선상에서의 복제기술

고가용성의 분야에서 시스템 이중화에 대한 내용을 언급했었는데, 복제기술은 그 연장선상에 있으며, Crash Recovery를 위한 가장 치밀한 전략적 방법으로 사용될 수 있다. 이 장에서는 복제와 그것이 가용성(Availability)에 미치는 영향에 대해서 웹 서버에 사용되는 단순한 파일 시스템 복제에서부터 실시간 거래 시스템에 사용되는 엄청난 규모의 데이터베이스의 트랜잭션 복제(Transactional Replication)를 가지고 설명할 것이다. ("프로세스의 복제"와 "Failover 시간 단축을 위한 복제" 기술도 언급하지만, 주로 "재난 복구(Disaster Recovery)의 용이함을 위한 복제" 기술을 다루고자 한다.)

복제기술이란 데이터를 하나의 디스크 집합에서 다른 하나의 완전히 다른 여분의(Redundant) 디스크 집합으로 이동하는 것이다. 복제는 디스크 미러링(Disk Mirroring)과는 다른 기술인데, 미러링은 디스크들을 하나의 논리적 볼륨(Logical Volume)으로 구성하여 가용성을 향상시킨 것이고, 이에 반해 복제는 디스크들을 두 개의 완전히 독립적인 부분으로 인식하는 것이기 때문이다. 복제가 제대로 수행되면 데이터베이스의 복구에 사용되는 모든 물리적 종속파일 파일(완전한 데이터베이스)이 복제된 시스템에서도 존재하게 된다. 결국 복제는 일과성이 유지되는 두 개의 동일한 데이터 집합이며, 이상적으로는 이 두 개의 디스크가 물리적으로 다른 호스트에 연결되어 있는 서로 다른 위치에 있는 디스크를 의미하게 된다. 복제는 RAID-1이나, RAID-5보다 더 많은 디스크 공간이 필요하게되며, 이렇게 투자함으로써 다양한 운영상의 장애나 성능 손상(Performance Outages)을 더 빨리 복구할 수 있고, 장애 시간(Downtime)에 대한 예측 및 방어가 가능하다.

복제가 필수적으로 요구되는 환경

○ 로드 밸런싱이 필요한 서버

복제된 웹 서버들은 어느 호스트에서든 동일한 URL을 가지고 접근하여 동일한 파일 시스템을 Access함으로써 정확히 같은 내용을 보여주어야 한다. 복제 서버 형태로 확장성 있는(Scalable) 서버 팜(Farm)을 구성할 때는 적어도 어느 정도 수준의 파일 시스템 복제가 필요하다. Front-End 웹 서버 클라이언트들을 위해서는 대형 NFS서버를 사용할 수 있다. 그러나 아주 대형 팜(Farm)에서는 그런 NFS 서버들조차도 서로가 서로의 사본이 될 수 있다. 소프트웨어 개발 환경에서 일반적으로 사용되는 파일 시스템(개발 도구, 라이브러리, 문서들이 복제의 좋은 예이다)을 사용함으로써 NFS 서버를 사용하고 있는 모든 개발자들에게 빠른 속도로 Access할 수 있게 해 준다.

○ **짧은 장애 극복 시간이 필요한 환경**

시간이 돈과 직결괴고, 경제적 절대 가치가 커질수록 장애 복구를 위한 시간은 아주 짧아져야만 한다. Call을 라우팅(Routing)하거나 제공해 주는 통신 관련 응용프로그램은 몇 초 내에 장애로부터 정상 작동되어야 한다. 또한 매매 거래에서 주문을 전달하고 확인하는 서버는 장애 시 5초에서 10초 내에 복구될 수 있어야 한다. 따라서 응용프로그램과 그 프로세스들을 복제하는 방법만이 요구되는 시간 내에 복구가 완료될 수 있다는 것을 보장할 수 있다. 이런 용도의 서버는 대부분 메인 프레임 서버이거나 완벽한 Clustering이 구현되어 있어야 한다.

○ **데이터 손실 극복과 빠른 복구를 위치**

만약, 데이터베이스에서 데이터 불일치가 발생하여 테이프 장치로부터 장시간 백업을 내려 받아 전체 복구를 해야 한다면 어떻게 할 것인가? 이건이 온라인 서비스라면 장애 복구를 도와야 할 직원들이 장애 시간 내내 고객 불편 전화에 시달리게 된다. 시간과 업무의 정확성이 중요한, 다시 말해 복제가 필수적인 환경에서 데이터베이스나 파일 시스템의 손실을 막기 위해 복수 서버에 복수 개의 복사본을 만들기 위한 투자는 당연한 것이다. 또한 데이터베이스의 체크포인트(Checkpoint)나 리두 로그(Redo Log, 또는 Archive)를 사용하여 복구할 수 있다. 혹은 데이터 손실이 발생한 시점을 알 수 있다면 특정 시간대의 상태로 돌릴 수 있는 TSPITR(Point Intime Recovery)가 가능하다. (그러나, 다양한 장애에 대처하기 위해서는 전체 복제가 필수적이다.)

○ **WAN의 병목현상 해결**

NFS와 다른 여러 파일 시스템 액세스 기법들은 LAN환경에서 사용할 목적으로 디자인되어 왔다. WAN 환경에서, 파일 엑세스 프로토콜(File Access Protocol)은 빠른 참조를 할 때나 탐색의 용도로는 쓸 만하지만, 파일에 대한 주문형(On-Demand), 실시간 액세스(Real-Time Access) 용도로는 대기 시간(Latency)과 대역폭(Bandwidth) 때문에 쓰기가 힘들다. 만약 여러 곳에서 동일한 데이터를 액세스하고, 평균 대역폭이 낮으며 일반적인 라운드 트립 대기시간(Round Trip Latency)가 길다면, WAN에서의 병목 현상을 없애기 위하여 데이터베이스나 파일 시스템의 복제를 사용해야 할 것이다. 경험에 따르면, 평균 액세스 시간이 두 배로 늘어난 경우가 바로 이러한 경우이다. 이런 경우 해결 방법은 복제된 데이터베이스를 로컬에서 액세스하는 것이다.(분산 데이터베이스 환경 구축)

○ **일반적인 재난 복구**

데이터베이스를 날려버리는 것은 논리적인 재난에 속한다. 그러나 하드웨어 자체의 장애나, 화재, 홍수 같은 자연재해의 경우는 어떻게 할 것인가? 만약 전체 복제가 이루어진 사이트가 있어서 서비스할 준비가 되어 있다면, 테이프로부터 데이터를 내려 받아 시스템을 완전히 다시 설정하는 복구 절차를 밟지 않더라도, 빠른 시간 내에 서비스를 다시 가능하게

할 수 있다. 데이터 손실이나 자연재해에 의한 복구 방법은 동일하며, 어느 경우든지, 최소한의 조치를 취하기 위해서는 응용프로그램 개발자와의 긴밀한 협조가 필요하다는 것을 알게 될 것이다. 재난은 복제가 가장 이상적으로 적용될 수 있는 시스템 수준의 재난과 데이터와 환경 요소에 대한 복제가 요구되는 환경적 재난의 두 가지로 나뉜다.

복제는 간단하게는 하나의 디스크를 완전하게 백업해서 정상적인 다른 시스템에서 다시 로드해서 사용하는 것도 있고, 복잡하게는 시스템에 로드가 심하게 걸리거나 장애 시 네트워크 상에 있는 다른 복제 시스템에 프로세서의 현재 메모리 상태를 복제하는 것까지를 포함한다. 복잡한 것에서 간단한 것에 이르기까지 복제 기법을 크게 5가지 목록으로 분류한다면 다음과 같다.

복제 기법

□ User에 의한 파일 시스템 복사

원거리 복제본을 만드는 가장 쉬운 방법임과 동시에 복제가 실패하거나, 장애로 인한 문제가 생길 소지가 가장 많은 방법이기도 하다. 파일 시스템 복사 기법에는 ftp, tar, dump, rdist 등이 이고, 마스터 서버에서 복수의 Slave로 파일을 전송하기 위해 특별한 백업 유틸리티를 사용하는 경우도 있다.

❷ Device Driver 수준의 쓰기 전송

이 방법은 자동화된 쓰기 복제 기법 중 가장 Low Level의 기법이다. 복제 드라이버가 설치된 서버에서 디스크 쓰기 작업이 이루어지면, 쓰기 명령은 원거리 복사가 끝날 때까지 전체 쓰기 명령이 블록킹된 상태로 다른 시스템에 복사된다.

❸ Disk Block 단위 복제

이 기법은 호스트 쪽의 복제 드라이버와 비슷하게, 하드웨어 디스크 어레이를 이용하여 변경된 데이터를 다른 디스크 어레이에 전송하는 기법이다. 바로 전 기법과 유사한 방법으로 한 번에 하나의 시스크 블록을 복제하게 되는데, 쓰기, 변경, 추가 작업이 많은 곳에서는 네트워크에 심각한 성능 저하를 가져올 수 있다.

❹ Transaction 단위 복제

이 방법은 하나의 데이터베이스 서버로 들어오는 트랜잭션을 복수의 데이터베이스 서버로 분산시킨다. 몇몇 데이터베이스 용용 프로그램은 한 번의 트랜잭션으로 2단계 커밋(Two Phase Commit)을 이용하여 두 데이터베이스 시스템에 트랜잭션을 동시에 적용하게 할 수도 있습니다. 트랜잭션 단위의 복제를 구현하는 또 다른 방법으로는 트랜잭션 프로세서 모

니터(TP Monitor)가 있고 비동기 큐 시스템(Asynchronous Queuing System)이 있다.

❺ Processor 수준의 Memory 상태 정보 복제

위 네 가지 복제 기술은 고정된 영역을 여러 곳으로 복제하기 위한 방법들이다. 업데이트가 자주 발생하고 데이터 이동량이 많은 환경에서는 메모리 내에 있는 자료를 다중 서버에 빈번하게 업데이트해야 한다. 프로세서 수준의 상태 정보 복제 기술은 이러한 경우에 업데이트된 정보가 모든 관련된 응용프로그램에 전달될 수 있도록 해 준다. 이 기법에는 체크포인팅(Checkpointing) 기법이 포함되는데, 이것을 이용하면 응용프로그램이 중요한 내부 상태 정보를 디스크에 저장해 두었다가, 장애 이후 혹은 업데이트 처리가 빈번하지 않은 시간에 다시 작업을 할 수 있게 해 준다.

데이터베이스 복제

복제기법들이 적용된 시스템 환경 중에서 데이터베이스 분야는 가장 광범위하게 복제 기술이 적용된 사례이며, 최근의 긴급 재난 복구(Disaster Recovery) 센터 구축 시 가장 많은 기술적 분야를 고려해야 하는 부분이기도 하다. 데이터베이스 복제는 다양한 재난, 예를 들어 주 데이터베이스가 손상되거나 동작 중이던 시스템의 장애시 MTTR을 분 단위가 아닌 초 단위로 낮추어 주고, 전체적인 시스템의 물리적인 장애로부터 시스템을 보호해 준다. 데이터베이스의 복제는 파일 복제 기법을 변형하거나 로그 재생(Redo Log Replay)을 이용할 수도 있고, 데이터베이스의 자체 기능, 예를 들면, 분산 트랜잭션 관리 혹은 써드 파티 트랜잭션 프로세싱과 큐 시스템을 이용하여 트랜잭션이 복수 서버에서 동작하는 복수 데이터베이스 인스턴스에 트랜잭션을 수행할 수 있게 해 준다.

데이터베이스 로그 재생은 데이터베이스 복제의 가장 단순한 방법이며, 파일 시스템 복제를 본딴 방법이다. 모든 데이터베이스는 특정한 구조, 일반적으로 “Undo"와 ”Redo"에 바뀐 내용을 저장하여 시스템 장애 발생시 트랜잭션을 다시 돌리는 데 사용한다. 로그 재생은 로그 파일을 복사해서 다른 시스템에 복사하고, 그리고 나서 복제 서버의 데이터베이스에서 복구 메니져(Recovery Manager)를 사용하여 그 트랜잭션을 다시 적용함으로써, 시간 간격은 좀 있지만 원본과 거의 동일한 데이터베이스를 만든다. 데이터베이스 복제 기술 중 하나인 Replication 기술에 대해서 알아보자

SYMMETRIC REPLICATION

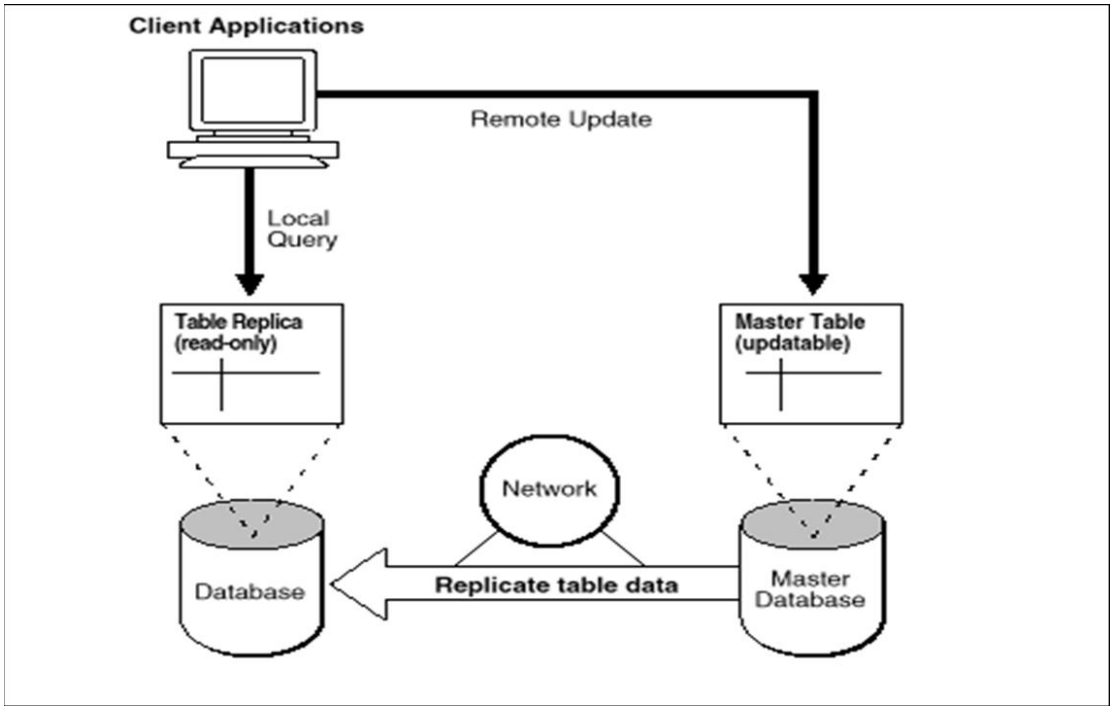
(이 환경은 Oracle RDBMS를 중심으로 설명하도록 하겠다)

분산 환경에서 다른 site에 있는 data를 local site에도 유지하면, network을 거치지 않으므

로 data를 access하는 순간의 performance도 향상되고 network failure가 발생하여도 작업을 계속 진행할 수 있다. 이렇게 분산 환경에서 다른 site에 있는 data를 자신의 site에 object로 복사하여 사용하는 것을 replication이라고 하며, Oracle에서는 다음과 같은 두 가지의 replication 방법을 제공한다.

Read-only snapshot(Basic Replication)

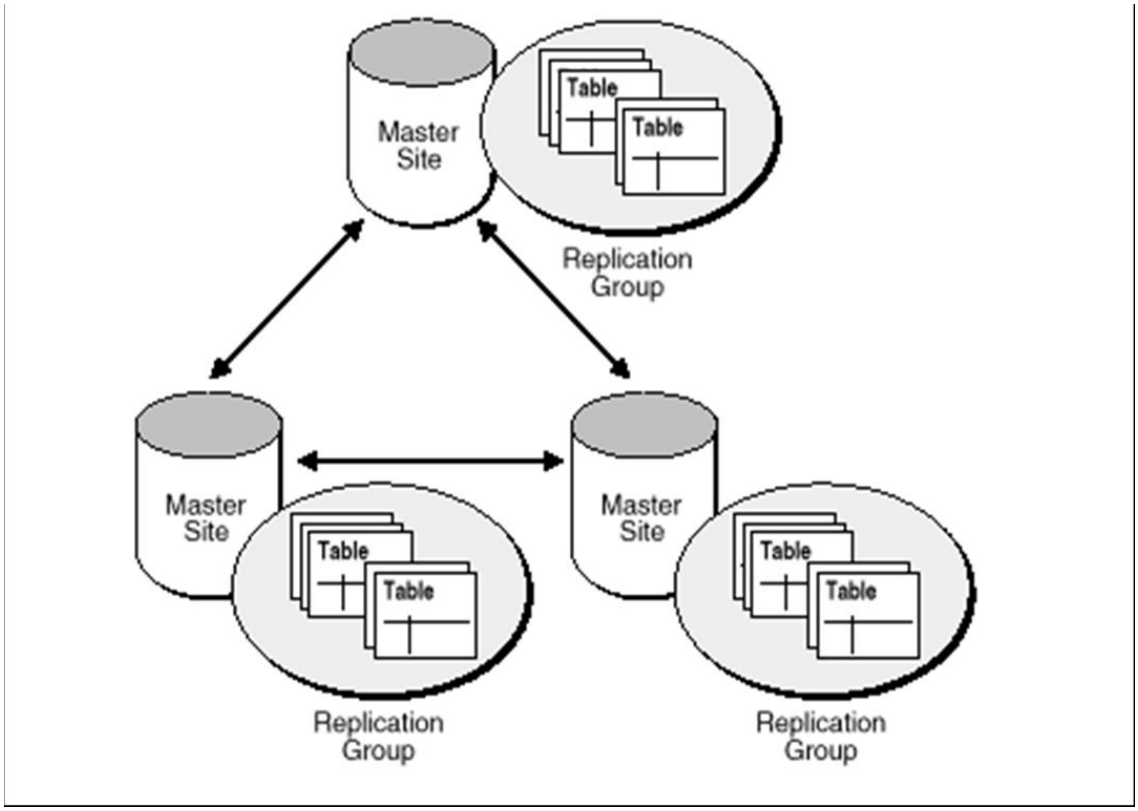
이것은 기본적인 replication 방법으로, 분산 환경 중의 한 node에 있는 master table에 대한 read-only snapshot을 local site에 유지하고 주기적으로 master table의 변경된 내용을 반영하게 된다.



[그림 5.1] Read-Only Snapshot

Symmetric replication(Advanced Replication)

read-only snapshot이 master table에서만 변경 작업이 가능하고 snapshot은 읽기만 가능한 데 반해 symmetric replication 환경에서는 이 환경에 포함된 모든 replicate된 data들을 변경 가능하고 이 변경 사항이 나머지 모든 site에 전달된다. 또한, read-only snapshot이 data-level의 변경 사항을 반영(propagate)하는데 반해 symmetric replication은 schema-level의 변경 사항도 다른 site에 반영할 수 있다.



[그림 5.2] Multi-Master Snapshot

asynchronous와 synchronous replication 기법의 비교

master site나 updatable site에서 다른 master site로 데이타를 replication하는 방법으로 asynchronous 방법과 synchronous 방법, 두 가지가 있다. 이 두 가지 방법을 비교하면 다음과 같다.

(1) asynchronous replication

- store and forward 기법

모든 트랜잭션은 deferred RPC queue에 일단 저장된 후 정해진 시간 간격마다 전달(propagate)된다. replicated master table을 포함하고 있는 어느 한 site가 문제가 생긴다하더라도, 나머지 site들 사이에서의 propagation에 전혀 지장을 주지 않는다. deferred transaction의 변경을 전달하는 시간 간격은 사용자가 원하는 대로 지정할 수 있다. 다른 site에서의 변화가 바로 local site로 반영되는 것이 아니므로 replication되는 table들 사이에 일시적인 불일치 상태가 발생하게 된다. 여러 site에 같은 데이타를 변경하는 경우 conflict가 발생하는데, 이 conflict는 변경 사항이 반영되기 전까지는 감지되지 않는다. conflict resolution 방법은 오라클이 제공하는 기본적인 방법과 사용자가 정의하는 방법이 있는데, 이러한 방법을 적용하기 위해서는 테이블에 필요한 정보를 저장할 컬럼을 추가해야 하는 등, replication 환경을 구축하기 전에 고려하여 구성해야 하는 사항이 있다. 그러므로,

replication 환경을 기반으로 어플리케이션을 구현하기 이전에 반드시 conflict resolution에 대해 결정하여야 한다. synchronous replication에 비해 response time이 빠르다.

(2) synchronous replication

다른 site의 변경사항이 즉시 local site에 반영된다. 같은 data가 여러 site에서 변경된다 할지라도, conflict가 발생하지 않는다. 변경 사항을 synchronous하게 반영하여야 하는 site가 down되었거나 network failure가 발생하면, network failure가 해결되거나 문제의 site를 replication환경에서 제거할 때까지 replication환경에 포함되어 있는 local site의 데이타에 대한 변경 작업은 수행될 수 없다. response time이 asynchronous replication에 비해 느리다.

Read-only snapshot과 Updatable snapshot의 비교

read-only snapshot은 query만 할 수 있고, master table만이 변경되어질 수 있다. Updatable snapshot은 remote master table 뿐 아니라 snapshot도 직접 변경할 수 있다. Updatable snapshot은 변경 내용이 master table로 전달된다. Read-only snapshot과 달리, updatable snapshot은 단일 master table로부터 만들어져야 하며, table의 join으로 구성되어질 수 없다. Updatable snapshot은 read-only snapshot과 마찬가지로 master table의 full copy 본을 유지할 수도 있고 master table의 row들 중 일정 조건을 만족하는 row들만을 선택한 것일 수도 있다.

Updatable snapshot과 replicated master의 비교

symmetric replication을 구성하는 두 가지 방법인 replicated master와 updatable snapshot의 차이점은 다음과 같다.

(1) updatable snapshot

"pull" technology, 즉, 필요한 때에 replication을 받아오는 기법이다. 이렇게 변경된 데이타를 받아오는 것을 REFRESH라고 한다. snapshot은 master table data의 subset만을 replicate하도록 지정할 수 있다. snapshot은 set-oriented로, 보다 긴 time interval 후에 여러개의 트랜잭션에 의해 이루어진 변경 사항을 보다 효율적으로, batch 성 형태로 replicate하게 된다. master에서 snapshot으로의 성능이 snapshot에서 master로의 성능보다 중요한 경우에, 그리고 snapshot에서 발생하는 트랜잭션의 수가 master에서 발생하는 트랜잭션의 수보다 훨씬 작은 경우에 적당하다. node가 replication으로 분리 가능하거나 ,master가 되는 한 node에만 connection되기를 바라는 경우에 적당하다. snapshot에는 unique index를 만들 수 없다. 즉 unique나 primary key constraint도 부여해서는 안된다. performance를 위해 index가 필요할 때는 unique가 아닌 일반 index를 생성하면 된다.

updatable snapshot의 경우는 simple snapshot만이 가능하기 때문에 대부분 fast refresh 방법을 사용한다. fast refresh 방식의 경우 master table의 변경 사항만이 log에 기록되어 snapshot site로 전달되는데, 이 때 performance상의 이유로 인해, log의 내용이 순차적으로 snapshot base table에 반영되지 않기 때문에 순간적으로 duplicate 상태가 될 수 있다. (이 문제는 ORACLE8에서 declarative constraints 기능을 이용하여, transaction이 끝난 후 constraint를 check함으로써 해결되어질 수 있다.)

(2) replicated master

"push" technology, 즉 replication을 전달해주는 방식이다. 이렇게 데이타의 변경 사항을 전달하는 것을 PROPAGATION이라 한다. replicated master는 replicate되는 테이블의 전체 데이타를 포함해야 한다. multi-master replication은 각각의 트랜잭션이 발생함에 따라 그 트랜잭션을 전달하여 replicate되도록 하는 방법이다. master site는 다른 master site와 replication 환경 내의 object 구성 등의 구조가 같기 때문에 failover system으로 사용 가능하다. 그러나, snapshot은 master와는 다른 내부 구조를 가지므로 failover system으로 사용하기에는 부적합하다. master site는 replicate되는 테이블 사이에 ROWID가 같지 않다. snapshot은 ROWID를 이용하여 refresh하지만, master는 primary key를 이용하여 propagate 한다. conflict가 발생하면, master site에서 감지하고 해결한다. 모든 master node들이 서로 연결되어 있어야 한다. replicated 되는 모든 table은 반드시 primary key를 가져야 한다.

지금까지 복제기술에 대한 사례로 데이터베이스 시스템의 Replication 에 대해서 알아보았다. 마지막으로 근래에 이슈가 되어온 재해 복구 시스템에 대해서 알아보자.

재해 복구 시스템

사실 대부분의 사람들은 실제 재해에 대해서는 이야기하는 것을 꺼린다. 재해는 인재(테러나 전쟁 등으로 인한)이든 천재(홍수, 지진 등)이든 간에 사무실에서의 대비 수준으로 해결할 수 있는 이상의 변화를 초래할 수 있다. 극단적인 경우 재산 손실뿐만 아니라, 인명 손실까지 발생할 수 있는 것이다. 이 때문에 실제로는 거의 대비할 수조차 없는 일들에 대해 준비해야 하고, 모든 사람들이 얘기하는 것조차 꺼려하는 이슈에 관해서도 토의를 해야 한다.

(항상 강조하지만, 두 가지 문제를 하나의 해법으로 대처하려고 해서는 안된다는 것이다. 언제나 그런 것은 아니지만, 고가용성과 재해 복구를 다른 접근 방법을 가지고 받아들일 필요가 있다. 즉, 고가용성을 위한 “장애 복구”와 “재해복구”에 대해서는 다른 이해의 관점을 가질 필요가 있다.)

재해복구 시스템 구성 시에는 다음과 같은 내용들을 고려해야 한다.

[박스 5.1] 재해복구 전략 수립시 고려사항

❶ RSO (Recovery Scope Objective)
- ▶ 계정계, 정보계, 대외계, ..
- ▶ 원격지 단순 데이터 백업,
- ▶ 재해대비 시스템 복구를 위한 백업

❷ RTO (Recovery Time Objective)
- ▶ 2시간내, 4시간내, 8시간내, 24시간내, ..

❸ RPO (Recovery Point Objective)
- ▶ 특정 백업 시점 데이터 복구
- ▶ 전일 마감 데이터 백업 시점
- ▶ 재해발생 시점 데이터 복구

❹ RCO (Recovery Communications Objective)
- ▶ 네트워크 복구 수준
- ▶ 지역 모점, 주요 영업점, 전 영업점, ..

❺ BCO (Backup Center Objective)
- ▶ 자체 2nd 센터에 재해복구시스템 구축
- ▶ 자체 2nd 센터에 전문업체와 재해복구시스템 구축 공조
- ▶ 재해복구시스템 구축 전문업체에 위탁

어떤 Business Process가 Critical 한가? 이 Critical Business Process가 얼마나 긴 시간동안 다운되어 있는가? financial loss는 어느 정도 발생하며, 견딜 수 있는가? 다운타임 동안 제공할 수 있는 최소한의 Service는 어느 정도 Level인가? 등등.... 이러한 분석 정보를 배경으로

첫째, **RTO(Recovery Time Objective)** : 몇 시간 안에 복구할 수 있는가?

둘째, **RPO(Recovery Point Objective)** : 어느 시점까지의 데이터로 복귀할 것인가?

셋째, **Dependency** : 업무간 상호 연관성은 어느 정도인가?

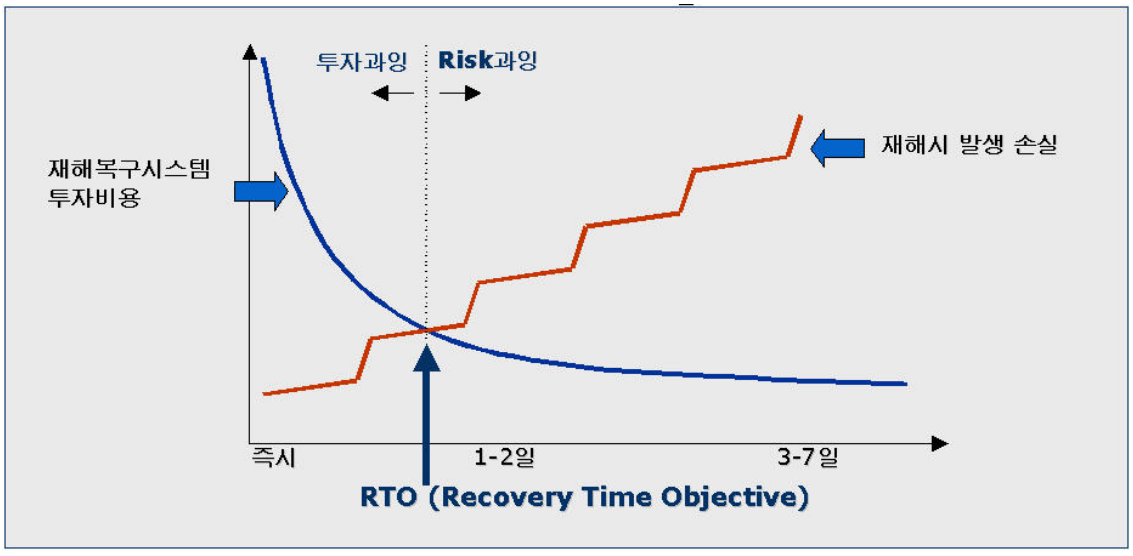
이 세 가지를 명확하게 판단하여야 한다.

또한 백업방법별로 그 특성을 고려하여 재해복구 센터를 구성할 때 반영하여야 한다. (아래 표 1.1 참조)

[표 5.1] 백업 방안별 특성

백 업 방 안	특 성
시스템 미러링	● CPU/Data Base Mirroring 으로 주센터와 백업센터간 동일한 시스템 Image 구성 ● 데이터 손실이 없어 재해/장애시에도 영향이 없는 시스템 구성 ● 영업점과 주센터 및 백업센터간 통신 라인 이중화 구성을 통해 신속한 복구 가능
실시간 데이터 이중화	● 실시간으로 주센터와 백업센터간 Data Base Level 의 이중화 백업체제 구성 ● 주센터와 백업센터간의 Data Base 를 직접 이중화하는 방안 - Log 또는 Journal 을 이용하여 주센터의 DB 를 백업센터에서 생성하는 방안 백업센터에서 LOG 를 적용하여 DB 를 갱신하는 시간이 소요(Dedicated DB 용량)
실시간 트랜잭션 로그 이중화	● 실시간으로 주센터와 백업센터간 Log 또는 Journal 만을 이중화 백업체제 구성 ● Data Base 는 주기적으로 원격 백업 또는 PTAM(Pickup Truck Access Method) ● 재발생시에 이중화된 Log 를 이용하여 백업센터의 Data Base 를 재해시점까지 복구
백업 테이프 이용 복구 방안	● 주기적으로 시스템 및 데이터 백업 테이프를 원격지에 PTAM 방식으로 소산 보관 ● 복구시스템 사용 계약을 통해 주기적 복구 시스템 구축 및 테스트 실시로 복구시간 단축 ● 점진적인 재해발생 대비 복구방안으로 급진적인 재해시 갱신된 데이터 손실

재해복구 사이트 자체에는 막대한 예산이 들어간다. 메인 사이트의 미러링이며, 상대적으로 메인 사이트와 떨어져 있기 때문이다. 이는 데이터 센터용의 공간, 사용자들을 위한 작업용 공간, 컴퓨터, 각종 집기류, 보안 설비, 그리고 이 밖에도 메인 사이트에 있었던 수많은 고가의 자재들이 동일하게 비치되어 있어야 함을 의미한다. 재해 복구 사이트에 별도 전문 인력을 파견하여 재해 발생시를 대비해 복구 사이트를 항상 유지하거나 보수 관리하는데 전력을 다하도록 하여야 한다. 고려된 RTO는 결국 투자비용에 반비례하며, 재해 시 발생 손실에 비례하기 마련이다.



[그림 5.3] DR 구성 시 Trade-Off 그래프

비용 문제는 그렇다 치더라도, 많은 수의 재해 복구 사이트는 대부분의 시간을 아무 것도 하지 않는 상태로 보내기 마련이다. 이 경우 메인 사이트에서는 자원 부족 현상이 빚어지고 있는 상황에서 막대한 양의 컴퓨팅 자원이 거의 놀고 있는 상태가 되면 구매 승인 담당자 측에서 볼 때 매우 신경에 거슬릴 것이다. 물론, 재해 복구 사이트를 마련하는 일은 텅 빈 집채만한 건물에 컴퓨터나 네트워크 다위를 잔뜩 갖다 놓고 재해가 발생하기만을 기다리면 되는 것을 훨씬 상회하는 성격의 것이다. 작업적인 측면에서의 요구사항을 마련하는 단계에서는 그 쏟아져 나오는 막대한 유형과 양의 작업들이 매우 복잡한 상호 연관성을 갖고 있다는 사실을 깨닫게 될 것이다. 재해 복구 시스템은 메인 사이트와 반드시 동기화되어 있어야 하며 이것은 하드웨어 수준에서 달성되어야 한다.

주 센터와 백업 센터와의 DR 구성 시, 동기화의 백업 병목지점을 판단하고 그 성능을 고려하여야 한다. 지역적으로 고려되어야 하는 요소는 “테이프 드라이브에 따른 용량과 성능“ , 그리고 ”데이터 버스에 따른 성능“이다. (DLT, AIT 장비는 수십 Gb에서 수 백 Gb를 동시에 백업처리할 수 있으며, 초당 성능 또한 다양하다. 데이터 버스 역시, Narrow SCSI에서 Wide, Ultra SCSI에 걸쳐, Fiber Channel 까지 다양한 대역폭(36Gb/Hour ~ 360Gb/Hour)을 제공한다. 그러나, 가장 빈번하게 병목요소가 되며, DR 구성 시 문제의 핵심 요소가 되는 요인은 Network 대역폭의 문제이다.(표 1.2 참조)

[표 5.2] DR 구성시 네트워크 대역폭의 고려

통신회선	대역폭	초당 전송량	사용용도
T1	1.5 Mbps	0.15 Mb/s	소량의 트랜잭션
T3	45 Mbps	4.5 Mb/s	중간급의 트랜잭션
ATM OC-3	155 Mbps	15.5 Mb/s	중간급의 트랜잭션
ESCON	200 Mbps	17 Mb/s	근거리 구성
Fiber	1024 Mbps	100 Mb/s	대량의 트랜잭션
DWDM	2.5 Gbps	2.5 Gb/s	대량의 트랜잭션

데이터가 재해 복구 센터로 옮겨진 후에는...

어떤 방법을 사용했던 간에, 일단 데이터가 원격 사이트로 옮겨진 후에는 해결되어야 할 이슈들이 많다.

첫째, 백업 테이프가 재해 복구 사이트에서 읽혀질 수 있는가? 읽어 들이기 위해 특화된 소프트웨어가 필요한 것은 아닌가? 소프트웨어 라이센스를 요구하는 것은 아닌가? 라이센스 키 등은 가지고 있는가? 라이센스 키를 입수하는 데 시간이 얼마나 걸리나? ..

둘째, 데이터 복원을 시작하기 전에 테이프 인덱스를 재구성해야 하는가? 얼마나 걸리나?

셋째, 실행 가능한 응용 프로그램들이 재해 복구 사이트에서 갖추어져 있는가? 가장 최신 버전인가? 사이트별로 실행하는 내용에 호환되지 않는 부분이 있는가? 관련된 데이터가 재

해 복구 사이트의 응용 프로그램 버전에서 사용 가능한가?
넷째, 응용 프로그램이 최신 버전으로 갖추어져 있다고 한다면, 재해 복구 사이트에서 실행하기 위해서 별도의 라이센스 키가 필요하지 않은가?

우선순위 정하기

재해 발생이 확인된 다음에는 시스템 관리자가 핵심 응용 프로그램을 재해 복구 사이트로 이전하여 필요한 모든 절차를 빠짐없이 수행해야 한다. 이들 절차는 자연스럽게 이어지되 빠르게 수행될 수 있으면 좋겠지만, 현실적으로는 그렇게 수월하지 않다. 필요한 사항들을 절차와 우선순위에 맞게 나열하여 보자
첫째, 어떤 프로그램이 먼저 실행되어야 하는가? 어떤 데이터가 먼저 복원되어야 하는가?
둘째, 재해 복구 사이트에서 전혀 필요하지 않은 응용프로그램은 무엇이 있는가?
셋째, 재해가 계속되는 경우 서비스를 대체할 어떤 프로그램이 필요한가?
넷째, 희소 자원에 대한 우선 순위는 누가 갖는가?
다섯째, 특정 그룹이 다른 그룹에 앞서 서비스를 받을 수 있도록 하는 결정은 어떤 근거로 내려지는가? (최고 결정권자가 없다면 대체 결정권자가 자리에 있어야 한다.)

전체 테스트

가장 중요하면서 간과할 수 없는 것이 재해 복구 절차의 전체 테스트이다. 재해 복구 절차를 모두 테스트하는 것은 실제 재해가 발생했을 때 가능한 한 자연스럽고 껄끄럽지 않게 실행에 옮길 수 있는 가장 좋은 방법이다. 모두 테스트하지 않게 되면 재해가 발생했을 경우 분명히 상황이 더욱 복잡 미묘하게 악활 될 가능성을 내포하게 되는 것이다. 주말이나, 연휴기간을 택해서 관련된 사용자 및 운용자들이 모두 참석한 가운데 실제 재해상황을 완벽하게 구성하여 놓고 모든 시나리오와 재해 단계별(Level별) 복구 테스트를 실시하여야 한다. 최악의 경우에서부터 Crash Level이 낮은 단계까지 절차적으로 우선순위에 따라 진행하여야 하며, 각 단계별 시나리오 별 복구 시간을 기록하여 복구 계획에 역으로 반영하여야 한다.
재해를 테스트하는 일은 대단히 비용이 많이 들어가는 일이지만, 그렇다고 테스트를 하지 않았다가는 사업을 완전히 접을 수(?) 밖에 없는 막대한 대가를 치를 수도 있으므로, 분명히 해야만 하는 필수 절차이다. 테스트 되지 않는 재해 복구 센터를 운영한다는 것은 재해 복구를 위한 모든 노력과 비용이 수포가 될 수 있는 시한폭탄을 끌어안고 있는 것과 같다.

6. 서비스 엔지니어의 관점에서 바라보는 ITIL의 적용

ITIL 서론

기업의 IT Infrastructure 관리의 초점이 IT 시스템에서 비즈니스로 옮겨가고 있다. 이전의 IT 인프라 관리가 IT시스템 자체의 성능에 집중되었다면, 최근에는 IT시스템을 비즈니스와 연계해 관리, 운영할 수 있는 비즈니스 관리 측면이 크게 부각되고 있다. 즉, IT 인프라 관리의 개념이 서비스와 비즈니스 가치를 포괄하는 비 IT적 측면으로 확대되고 있는 것이다.

지금까지의 IT 인프라 관리는, 네트워크, 애플리케이션, 시스템, 데이터베이스 등 기업의 IT 인프라를 구성하는 요소들의 성능과 안정성 등 IT적 관점에서만 관리된 경향이 강했기 때문에 IT시스템이 실제 비즈니스에 미치는 영향을 체계적으로 분석하는데 어려움을 겪었으며, 성공적으로 구축된 IT 시스템이라 할지라도 ROI를 산출하는 데 취약하다는 지적을 받아 왔다. 그러나 비즈니스 측면에서 IT인프라를 관리한다면 IT시스템이 어떤 비즈니스를 수행하고 창출하는지를 정확하게 정의하고, 전산 장애가 발생했을 경우 비즈니스에 어떤 영향을 미치는지를 파악해 대처할 수 있게 된다. 이를 통해 기업들은 IT 인프라 관리 비용을 절감하고 비즈니스에 최적화된 IT 인프라를 유지할 수 있게 된다.

지금까지 기업 내의 IT 관리부서나 CIO는 나름대로의 독자적인 경험에 의존해 IT 업무 프로세스 관리를 해오고, IT 운영을 위한 표준 절차나 가이드라인이 명확하지 않아 상당한 업무 비용 손실을 초래하고 질 높은 서비스를 제공하지 못했다. 최근 이 같은 문제들을 극복하기 위해 기업들은 보다 체계화되고 선진화된 표준 운영 프로세스인 ITIL(IT Infrastructure Library)을 도입해 IT 관리 운영에 대한 기본 틀을 정립하려고 노력 중이다. 최근 가트너가 발표한 자료에 의하면 현재 세계적으로 약 25%의 기업들이 ITIL을 활용하고 있고 2004년에는 ITIL 도입이 본격화되어 지금보다 약 5~10% 더 많은 기업들이 적극적으로 활용할 것으로 전망하고 있다.

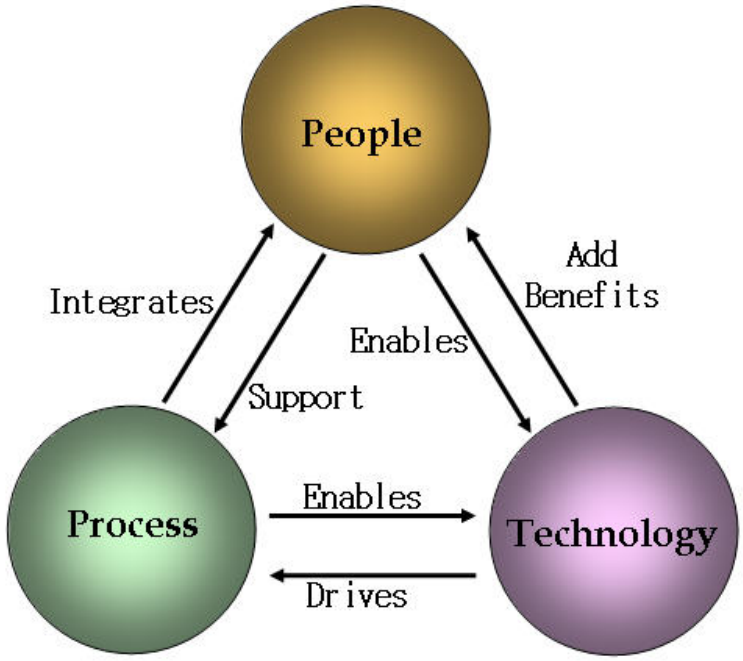
ITIL의 개념

IT Infrastructure Library를 의미하는 ITIL은, IT 서비스 관리(ITSM: IT Service Management) 에 대한 프레임워크 구현을 돕기 위한 문서들의 집합이다. ITIL 프레임워크는, 특정 기업 내의 복잡한 IT 환경에 대해 비지니스와 서비스 중심의 최적의 프레임워크를 제시한다. ITIL은 어떤 종류의 조직에도, 어떤 규모의 기업에도 활용 가능하고, 어떤 벤더에도 종속적이지 않은 포괄적이면서도 공개적인 표준 가이드로서, 1986년 영국 정부(CCTA : Central Computer and Telecommunications Agency)의 의해 개발되어 현재 업계 선두 10000개 기업들로부터 그 유효성과 효율성을 검증받아 표준 IT 서비스 관리 프레임워크로

사용되고 있으며 고객에게 고품질의 IT 서비스를 제공할 수 있는 기반으로 부각되고 있다. IT 프로세스 관리 프레임워크 Best Practice를 기반으로 만들어진 ITIL은, 제공하는 프로세스 및 서비스에 따라 다음의 다섯 가지 영역으로 구성되며 서로 밀접한 관계와 인터페이스를 갖고 있다.

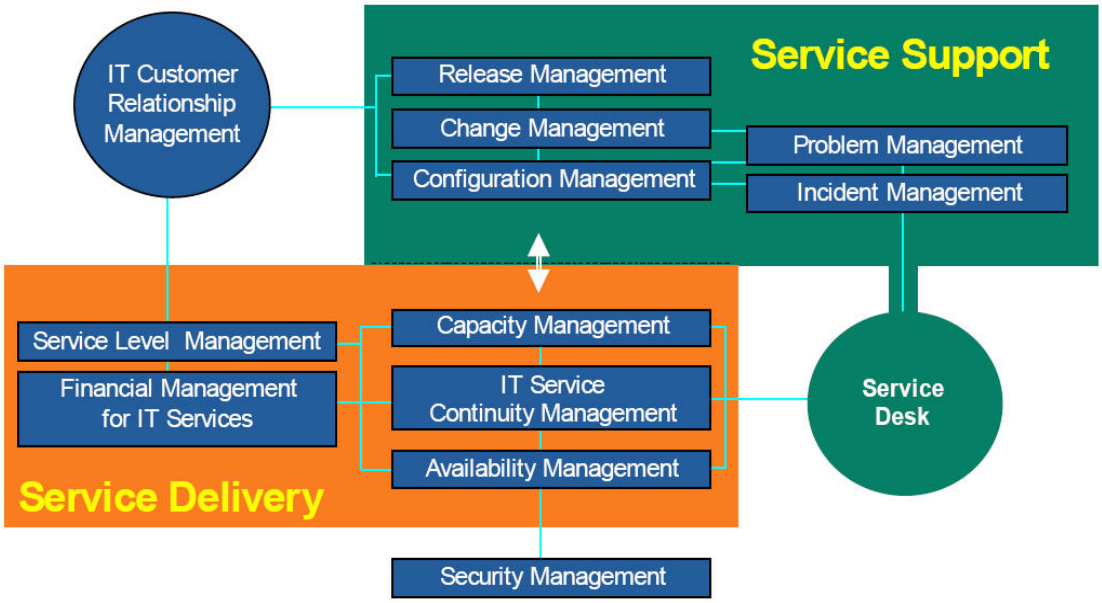
- **Service Delivery** : IT 서비스 제공자가 비지니스 고객에게 충족한 지원을 제공하기 위해 필요한 서비스 및 프로세스를 정의하고 있다.
- **Service Support** : IT 서비스 사용자가 비지니스 관련 IT 서비스를 항상 받을 수 있도록 보장하는 데에 필요한 관련 프로세스를 포함하고 있다.
- **Applications Management** : 소프트웨어 개발 라이프 사이클을 포함하고 있으며 소프트웨어 라이프 사이클 지원 및 IT 서비스 테스팅까지 확장하여 다루고 있다.
- **ICT Infrastructure Management** : IT Infrastructure를 운영 관리하기 위해 필요한 주요 프로세스를 포함하고 있다.
- **The Business Perspective** : 전체 비지니스의 중요 부분으로서의 IT 서비스 제공 품질의 개선 및 이해를 다루고 있다.

ITIL은 IT 서비스를 고객의 비지니스 요구사항과 연계하여 보다 나은 서비스를 제공할 수 있도록 IT 조직에 Best Practice를 제공한다. 즉, ITIL은 방법론이 아니라 IT 조직이 IT 서비스 관리를 할 수 있도록 제공되는 실천 모델이다.



[그림 5.3] People, Process, Technology에 의한 IT 서비스 관리

ITIL은 고품질의 IT 서비스 제공에 초점을 맞춰 IT 조직, 고객 및 파트너를 중심으로 다루고 있다. 여기서 IT 서비스라 함은 사람(People)에 의해 수행되고, 기술(Technology)에 의해 지원되는 프로세스(Process) 라고 할 수 있다. 즉, 서비스란 기술의 지원을 받은 조직에 의해 수행된 프로세스의 결과이고, 따라서 체계적으로 정의된 프로세스를 갖고 있지 못한 조직은 일도 제대로 수행할 수가 없다. 그러므로 체계적이지 못한 IT 운영 관리 프로세스는 반드시 정립되어야 하고, IT 운영 관리 프로세스 정립 및 혁신을 위해서는 전세계적으로 검증된 모델인 ITIL이 요구된다. 이처럼 People, Process, Technology 세가지 요소들의 최적화된 융합으로 IT 조직을 비지니스 기반의 운영 방식으로 바꾸어, 보다 우월한 IT 서비스를 제공하도록 하는 운영 방식이 IT Service Management (ITSM) 이다. ITSM은 운영자 중심이 아닌 고객과 프로세스 중심의 운영방식이라 할 수 있고, 협의된 서비스 수준1(Service Level Agreements) 에 맞는 서비스를 제공함으로써 효율적인 프로세스 운영을 통해 비용절감을 목표로 한다. Gartner: "Processes forms a big part of IT service management. Availability in a complex computing environment does not happen on its own or automatically by acquiring high-availability technology. It takes strategy, planning, policy and implementation to achieve it. These are people and process issues, not technology issues."



ITIL은 서비스 써포트(Service Support)와 서비스 딜리버리(Service Delivery)라는 핵심 영역으로 IT 프로세스 영역을 구분하고 있는데, 서비스 써포트는 IT 서비스 제공의 융통성과 안정성을 보장하기 위해 사건, 장애, 변경, 버전, 설정 관리로 구분되어 있고, 서비스 딜리버리는 IT 서비스의 품질과 비용 효율성을 보장하기 위해 서비스 수준, 가용성, 용량, IT

서비스의 재무, IT 서비스 연속성 관리로 구분되어 있다. 즉, 서비스 써포트는 IT 서비스 사용자가 고품질의 서비스를 제공받도록 하는 데에 필요한 관련 프로세스들을 포함하고, 서비스 딜리버리는 다양한 각도에서 비지니스 요구에 대한 분석을 하여 고객이 원하는 수준의 서비스를 정의, 계약하고, 정의된 서비스 수준을 위해 요구되는 요소 - H/W, S/W, 테크놀러지, 가용성 및 금전적 요구 조건- 들을 파악하여 서비스 수준을 모니터링하는 것이다. 또한, ITIL은 ISO 9000 품질시스템과 EFQM(European Foundation for Quality Management) 및 CMM(Capability Mature Model)과 깊은 관계를 갖고 있고 IT 서비스 관리에 필요한 주요 프로세스 및 Best Practice 를 제공함으로써 이러한 품질 시스템을 지원하고 있다. 따라서 기업들이 ITIL을 선택함으로써 IT 서비스 관리 분야에서 ISO와 같은 품질 시스템에 대한 인증을 공인받을 수 있고 IT 서비스에 대한 품질 및 비즈니스 중심의 IT 프로세스 실천 모델을 보유할 수 있게 될 것이다.

ITIL의 도입 효과

이미 유럽의 대다수 기업에서 1990년대 말에 ITIL을 적용했고 미국에서도 ITIL의 도입이 꾸준히 늘고 있다. 그렇다면 전세계적으로 많은 기업들과 정부기관이 ITIL을 적용하는 이유가 무엇일까 알아보자. 그리고 투자비용대비 이득 측면에서도 그 이점을 살펴보자.
첫번째로, 기업 내에서 얻는 효과로는 다음과 같다;

- IT 변경 사항에 대한 관리, 제어권의 향상
- IT 비용 관리 프로세스를 통한 IT 비용의 절감
- IT와 비즈니스를 연계 관리
- IT 관리도구의 적용을 위한 프로세스 Setup
- 아웃소싱에 대한 결정을 위한 프레임워크 준비
- 상호 대화를 위한 단일화된 참조모델
- 체계적/명확한 IT 조직체계
- 프로세스 표준화
- 일의 중복 감소
- ISO 9000 인증 가능

두번째로, IT 고객은 다음과 같은 효과를 기대할 수 있다;

- 문서로 상세하게 잘 정리된 IT 서비스 - 보다 안정적인 IT 운영환경
- 제공 서비스에 대한 품질 보증에 따라 신뢰성 증가
- 명확한 대화 통로 제공
- 신속한 신규 IT 서비스 Launch

세번째로, 신속한 ROI를 기대할 수 있다.

미국 내의 ITIL 컨설팅 회사인 Interprom 사에 의하면 ITIL을 적용한 고객사의 경우 IT downtime 을 65%를 줄임으로써 100명의 사용자 당 연간 197,000 USD의 절감 효과를 보았으며 7M USD의 수입 손실을 감소, 그리고 22일의 투자회수 기간이 산정되었다. 이 경우의 고객은 직접 ITIL을 지원하는 관리도구를 적용한 경우이다. 또 다른 고객의 경우, 서비스 콜의 해결 시간을 50% 감소시켰으며 1주일에 처리 가능한 콜 량도 450건에서 2000건으로 증대함으로써 생산성을 향상시켰다. ITIL이 전세계적으로 주목받고 있는 상황에서, IT 관련 서비스를 창출하고 제공하는 기업들은 자사뿐만 아니라 고객사 및 외부 기업의 IT 관리 조직이 세계적으로 검증된 표준이 될 만한 프로세스 모델인 ITIL을 채택하도록 도와주고, ITIL 과 더불어 CMM 등의 다른 프로세스 모델을 통합적으로 적용하는 방안에 대해서도 검토할 수 있어야 하겠다.

“파이팅 시스템 엔지니어!!“라는 표제어로 3개월간의 연재를 마치면서, 마지막으로 ITIL에 대한 시스템 엔지니어들이 관심을 권장하고 싶다. ITIL에 대하여 혹자는 한 때의 유행처럼 여길 수도 있고, 혹자는 컨설팅 업체의 상술로 비하하는 이도 있지만, ITSMF에서 추진하는 ITIL의 적용과정을 통하여 고객과 조직(엔지니어)과 상품(기술)이 어떤 프로세스로 흐르며, 어떤 메쏘돌로지와의 조화로운 통합적 적용 방안으로써 작용하는지 알고, 우리의(엔지니어의) 업무 체계를 폭넓게 이해하고, 능동적으로 이끌어 갈 수 이해력을 키울 수 있을 것이다.

참고자료

❶ 2002A Bluprints for High Availability by Evan Marcus and Hal Stern

❷ http://www.itil-itsm-world.com/

❸ http://www.itil.com/