

논문 2012-49-9-23

## 재해 복구를 위한 백업 시스템의 안정성 검증

( A Stability Verification of Backup System for Disaster Recovery )

이 문 구\*

( Moon-Goo Lee )

### 요 약

시스템의 장애와 재해로부터 기업의 자산을 보호하는 것은 IT 운영 관리자들이 가장 고려해야 할 사항들 중 하나이다. 그러므로 본 연구는 이러한 재해 복구를 위한 백업 시스템을 제안하였다. 기존의 백업 방식이 데이터베이스의 업데이트가 발생하면 그 이력이 개장 기록(redo log) 파일에 저장되고 이 기록파일이 예정된 파일의 크기를 넘어서면 순차적으로 기록보관소 기록(archive log) 파일들로 저장된다. 그러므로 데이터베이스의 변경이 생기는 동안 실시간 변경되는 데이터를 백업받는 과정에 데이터 손실의 오류가 발생할 수 있다. 제안한 백업 시스템은 온라인 개장 기록을 처리기록(transaction log) 데이터베이스로 실시간 백업하고, 기존 백업방법에서 누락 될 수 있는 자료보관소 기록(archive log)으로 백업된다. 그러므로 데이터의 복구 시 온라인으로 개장 기록까지 실시간 복구가 가능하여 데이터의 손실을 최소화 할 수 있을 뿐만 아니라, 멀티스레드 처리방법으로 데이터 복제가 수행되어 시스템의 성능도 향상 시킬 수 있도록 설계하였다. 설계된 백업 시스템의 안정성을 검증하기 위하여 CPN(Coloured Petri Nets)을 도입 하였으며, 백업시스템의 각 수행 단계는 CPN의 그래픽으로 도식화 하고, CPN의 정의와 정리를 기반으로 안정성을 검증하였다.

### Abstract

The main thing that IT operation managers consider is protecting assets of corporation from system failure and disaster. Therefore, this research proposed a backup system for a disaster recovery. Previous backup method is that if database update occurs, this record is saved in redo log, and if the size of record file is over than expected, this file is saved in archive log in order. Thus, it is possible to occur errors of data loss from the process of data backup which change in real time while changes of database occur. Suggested backup system is back redo log up to database of transaction log in real time, and back a record that can be omitted from previous backup method up to archive log. When recover the data, it is possible to recover redo log in real time online, and it minimizes data loss. Also, throughout multi thread processing method data recovery is performed and it is designed that system performance is improved. To verify stability of backup system CPN(Coloured Petri Net) is introduced, and each step of backup system is displayed in diagram form, and the stability is verified based on the definition and theorem of CPN.

**Keywords :** disaster recovery, redo log, transaction log, archive log, multi thread, stability verification

### I. 서 론

업무의 IT 의존도가 높아지면서 데이터 및 정보는

기업 활동의 원천이기 때문에 데이터 및 정보의 보호는 업무 연속성 측면에서 IT 운영의 최대 우선순위가 되고 있다. 그러므로 이러한 재해에 대비한 고가용성(high availability) 백업시스템을 설계하였으며, 이는 정보시스템의 장애로부터의 예상 복구소요시간이 허용 가능한 범위를 초과하여, 정상적인 업무 수행에 지장을 초래하는 피해에 대비한 재해 복구(Disaster Recovery) 시스템이다. 제안하는 백업시스템은 온라인 개장 기록(redo

\* 평생회원, 김포대학 IT 학부 인터넷정보과  
(Div. of IT, Dept. of Internet Information,  
Kimpo College)

※ 이 논문은 2012학년도 김포대학의 연구비 지원에 의하여 연구되었음.

접수일자: 2012년7월5일, 수정완료일: 2012년9월3일

log)을 처리기록(transaction log) 데이터베이스로 실시간 백업함으로써 기존 백업방법에서 누락 될 수 있는 자료보관소 기록(archive log)으로 백업한다. 그러므로 데이터의 실시간 복구가 가능하고 복구 시 온라인으로 개장 기록(redo log)까지 복구가 가능하여 데이터의 손실을 최소화 할 수 있다. 또한, 데이터베이스에서 같은 PK(primary key)에 대해서 변경되는 대상(target) 데이터들을 같은 스레드로 모아서 멀티스레드로 처리함으로써 실시간 변경되는 데이터에 대한 복제기능이 수행되어 시스템의 성능도 향상 시킬 수 있도록 설계하였으며, 제안하는 백업 시스템의 안정성을 검증하기 위하여 컬러페트리넷(CPNs)을 이용하였다.

본 논문의 구성에서 II장은 관련연구, III장은 CPN 도입목적과 검증방법, IV장은 제안하는 백업시스템의 안정성 검증, 마지막으로 V장은 결론과 차후 연구방향 등을 기술하였다.

## II. 관련 연구

### 1. 백업시스템에 관한 연구

기존의 데이터베이스 백업 방식은 백업되지 못한 기록보관소(archive) 기록(log) 파일의 손실, 온라인 개장 기록(redo log) 데이터의 손실, 그리고 실시간 처리량이 많은 서버의 경우 데이터의 손실률이 증가 하여 데이터의 복구 시 시간이 오래 걸릴 뿐만 아니라 실시간 복구의 어려움이 있다. 이러한 기존의 데이터베이스 백업 방식의 구성은 [그림 1]과 같다.

기존의 백업 방식은 운영 데이터베이스의 업데이트나 백업과정이 수행되면 그 이력이 개장 기록(redo log) 파일에 저장되고 이 기록파일이 어느 정도의 지정된 크기를 넘어서면 순차적으로 기록보관소 기록(archive log) 파일들로 스위칭을 하며 저장한다. 예를 들어 기록보관소 기록 파일의 크기를 4GB 단위로 지정을 하면 데이터베이스의 변경되는 이력이 개장 기록에 4GB까지 저장되면 기록보관소 기록의 인덱스를 하나 증가시켜 스위칭을 한다. 개장 기록 파일은 데이터베이스의 변경이 생기는 동안 계속 변경되기 때문에 기존 백업으로는 데이터를 받을 수 없다.

기존 백업방식의 문제를 해결하기 위해서 [그림 2]와 같이 하이브리드 백업 방식을 제안한다. 시스템의 구성은 온라인 개장 기록(redo log)을 처리기록(transaction

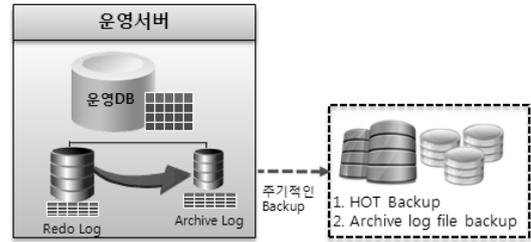


그림 1. 기존 백업 시스템의 구성  
Fig. 1. Configuration of the existing Backup System.

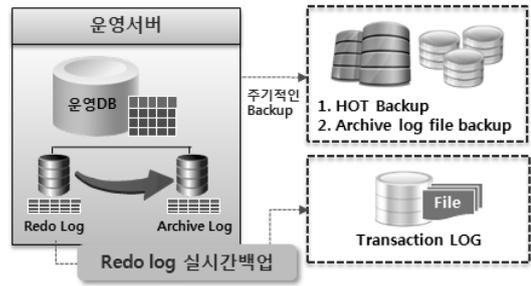


그림 2. 제안하는 백업 시스템의 구성  
Fig. 2. Configuration of Proposed Backup system.

log) 데이터베이스로 실시간 백업하고 기존 방식의 백업에서 누락 될 수 있는 자료보관소 기록(archive log)으로 백업한다. 그러므로 데이터의 실시간 복구가 가능하고 복구 시 온라인으로 개장 기록(redo log)까지 복구가 가능하므로 데이터의 손실을 최소화 할 수 있다. 이러한 개장기록의 복구를 위해 변경된 데이터를 반영하는 SQL 스크립트의 변환이 가능하고, 또한 데이터베이스에서 같은 PK(primary key)에 대해서 변경되는 데이터들을 같은 스레드로 모아 실행시키는 멀티스레드 대상 데이터베이스 적용 방식으로 실시간 변경되는 데이터에 대한 복제 성능도 향상 시킬 수 있다.

## III. CPN 도입목적과 검증방법

### 1. CPN의 도입 목적

재해복구를 위하여 제안하는 백업시스템의 안정성 검증을 위해서 CPN(Coloured Petri Nets)을 도입하게 된 주요 목적은 백업시스템의 각 기능에 대한 흐름이 CPN의 아크와 노드 등으로 구성된 그래픽으로 표현과 시스템의 각 진행단계를 논리형식에 맞는 구문의 표현으로 정의가 가능하며, CPN의 성질(안정성, 위계성, 보존성)을 이용하여 시스템을 구현하기 이전에 안정성을 검증할 수 있다<sup>[1]</sup>. CPN을 도입하여 시스템의 각 기능에

대한 동적 특성 중에 시스템의 교착상태(deadlock)의 발생 여부, 토큰간의 충돌 발생 여부 등과 관련하여 시스템의 안정성(safety)을 검증할 수 있다.

가. CPN 그래픽 표현방법

CPN의 표현방법은 [그림 3]과 같으며, 칼라셋(color set)은 토큰의 타입을 나타낸다. 아크식은 아크의 표현값을 나타내고, 가드(Guard)식은 전이의 조건을 나타낸다. 초기 마킹은 플레이스에서 표현되어, 초기의 토큰값을 나타내며, 선언노드는 현재 페이지에서 사용하는 칼라와 변수를 선언한다<sup>[6-9]</sup>.

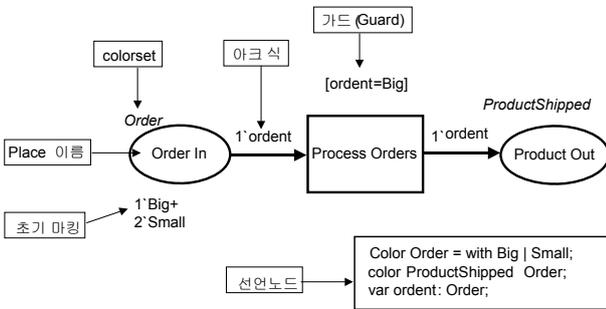


그림 3. CPN의 표현방법  
Fig. 3. Express of CPNs.

나. CPN의 정의와 검증방법

CPN은 다음의 요구사항을 만족하는 튜플(tuple)로 구성된다. [정의 1]

$$CPN = (\sum, P, T, A, N, C, G, E, I)$$

- $\sum$  : 0이 아닌 타입의 유한 집합 형태의 컬러 집합.
- $P = \{P_1, P_2, \dots, P_m\}$  : 상태(Place)의 유한 집합으로 어떤 사건이 발생하기 전이나 발생한 후의 상태.
- $T = \{t_1, t_2, \dots, t_n\}$  : 전이(Transition)의 유한집합으로 어떤 상태로 도달하기 위한 사건.

$A : A \subseteq (P \times T) \cup (T \times P)$ 를 나타내며, 흐름관계의 유한 집합으로, 전이(transition)의 흐름.

$N$  :  $N$ 은 노드(Node)의 함수  $N(a)$ . 만약 근원지(source)에서 목적지(destination)로 간다면,  $N=(source, dest)$ 를 표현 한다.

$G$  : 가드(Guard)의 함수

$$\forall t \in T :$$

$$[Type(G(t)) = B \wedge Type(Var(G(t))) \subseteq \sum]$$

$B$ 는 바인딩엘리먼트(binding element)  $b$ 의 유한집합.

$E$  : 간선 식 함수.  $C$  : 컬러(Color)의 함수.

$$\forall a \in A : [Type(E(a)) = C(p(a))_{MS} \wedge , p(a) \text{는 } N(a) \text{의 상태(place)이다.}$$

$I$  : 초기화 함수로서, 다음의 닫힌 식  $P$  로부터 정의.

$$\forall p \in P : [Type(I(p)) = C(p)_{MS}]$$

가드 함수  $G$ 는 변환  $t$ 와 대수형의 식 즉, 술어를 사상한다.  $G(t)$ 의 모든 변수는  $\sum$ 에 속하는 데이터 타입을 가져야만 한다.

모든 노드의 집합을 표시하기 위해서  $X = P \cup T$ 를 사용한다. 그리고 CPN 구조의 이웃 요소들 사이의 관계를 기술하는 많은 함수를 정의한다. 각 함수의 이름은 함수의 범위를 나타내며, P는 상태에 사상되고, A는 간선의 집합에 사상된다. 컬러집합은 CPN에서 사용되는 유형, 연산 함수들을 결정한다. 각 컬러집합은 적어도 한 개의 요소를 갖고 있다고 가정한다. 상태(P), 전이(T)와 간선(A)은 집합 P, T, A의 조합으로 정의된다. 상태, 전이, 그리고 간선을 갖는 집합은 유한해야 한다. 노드 함수는 첫 번 요소가 출발 노드이고, 두 번째가 목표 노드인 쌍을 각 간선으로 사상한다.

[정의 2]

가중치 함수(weight function)  $W$ 가 상태(place)의 흐름을 나타내기 위한 필요충분조건은  $W$ 가 상태(place)의 불변식(invariant)을 결정하는 (정리 1)과 (정리 2)를 만족하여야만 한다.

(정리 1) 다음의 조건을 만족하는 필요충분조건인 경우  $W$ 는 상태의 흐름(flow)이라고 말한다.

$$\forall (t, b) \in BE : \sum_{p \in P} W_p(E(p, t) < b >) = \sum_{p \in P} W_p(E(t, p) < b >). \tag{1}$$

$W_p$ 는 모든  $p \in P$ 에 대한 가중치의 집합이다.

$BE$ 는 모든 바인딩 엘리먼트(binding element)들의 집합이고, 바인딩 엘리먼트는  $(t, b)$ 의 쌍으로 이루어지며,  $t \in T$  그리고  $b \in B(t)$ 이다.

(정리 2) 다음의 조건을 만족하는 필요충분조건인 경우  $W$ 는 상태 불변식(place invariant)을 결정한다고 말한다:

$$\forall M \in [M_0 > :$$

$$\sum_{p \in P} W_p(M(p)) = \sum_{p \in P} W_p(M_0(p)). \quad (2)$$

W가 상태(place)의 흐름(flow)이라고 가정할 때,  $M_1 \succ M_2$  가  $W(M_1) = W(M_2)$ 임을 증명한다면 제안하는 모델의 안정성을 검증하게 된다<sup>[1, 3]</sup>.

#### IV. 제안하는 백업시스템의 안정성 검증

##### 1. 백업시스템의 구조

제안하는 백업시스템은 [그림 4]와 같이 백업을 위하여 각 데이터의 복제과정을 수행하도록 설계하였으므로, 근원지(source) 시스템의 복제 에이전트(Replicator Agent)와 대상(target) 시스템의 복제 에이전트(Replicator Agent)로 각각 구성된다<sup>[2]</sup>.

근원지 시스템의 복제 에이전트(Replicator Agent)는 파일 캡처 관리자(File Capture Manager)를 실행하는데 이는 파일시스템의 데이터에 대한 변경내역을 근원지 시스템에서 대상 시스템으로 이동하여 실시간 동기화할 수 있도록 설계하였다. 근원지 시스템 복제 에이전트(Replicator Agent)의 DBMS 캡처 관리자(DBMS Capture Manager)는 운영 중인 데이터베이스의 변경내역을 추출하여 근원지 시스템에서 대상 시스템으로 데이터를 전송하여 실시간 동기화할 수 있도록 설계하였다. 파일시스템의 동기화를 위한 파일 캡처 관리자(File Capture Manager)는 근원지 시스템에서 데몬 형태로 구동하며, 동기화의 대상이 되는 파일시스템의 변경내역을 주기적으로 감시하여 변경된 파일을 대상 시스템의 파일 공급 관리자(File Apply Manager)로 전송하여 반영하는 방식으로 동기화를 수행한다<sup>[5, 10]</sup>. 이때 변경된 파일 전체를 전송하지 않고, 블록단위의 체크섬(checksum)을 비교하여 변경된 블록만을 추출하여 압축전송하게 함으로써, 네트워크 부하를 최소화 할 수

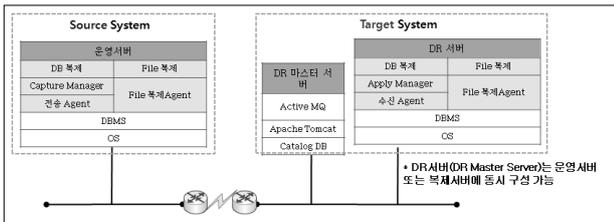


그림 4. 백업시스템의 구조  
Fig. 4. A Structure of Backup System.

있도록 설계하였다. 데이터베이스 동기화를 위한 DBMS 캡처 관리자(DBMS Capture Manager)는 근원지 시스템에서 데몬 형태로 구동하며, 근원지 데이터베이스의 변경내역을 개장 기록(redo log)으로부터 실시간으로 추출하여 대상 시스템으로 전송하고, 대상 시스템의 DBMS 공급 관리자(DBMS Apply Manager)는 전송된 데이터 변경내역을 최적화된 SQL 구문으로 복제 데이터베이스에 반영하도록 하여 고성능의 실시간 데이터베이스 동기화가 수행될 수 있도록 설계하였다. 복제 마스터와 콘솔(Replication Master & Console)은 동기화를 위한 모든 정책 설정 정보의 관리와 동기화 상태의 모니터링을 수행할 수 있는 관리 기능 및 모니터링 그래픽 사용자 인터페이스 기능을 수행한다<sup>[4]</sup>.

##### 2. 백업시스템의 CPN 표현

[그림 5]는 CPN을 이용하여 백업시스템의 안정성 검증을 위한 그래픽 표현으로, 최초로 임계값(critical value)을 가져오는 과정에서 오류를 체크한다.

##### 가. 백업시스템의 안정성 검증

백업시스템의 안정성 검증을 위한 CPN의 각 도달 가능한 상태들은 다음과 같이 4가지 도달 가능한 상태로 표현 할 수 있다. 각각의 경우는 상태 및 전이 불변식에 따라 같은 방법으로 그 안정성이 증명된다.

$$M_{Begin-rep} [Y_{Check-CV} Y_{Process-logging1} > M_{EOP} \quad (3)$$

$$M_{Begin-rep} [Y_{Check-NV} Y_{Check-CPMGR} Y_{Process-logging2} > M_{EOP} \quad (4)$$

$$M_{Begin-rep} [Y_{Check-NV} Y_{Check-CPMGR} Y_{Check-DPMGR} Y_{Process-logging2} > M_{EOP} \quad (5)$$

$$M_{Begin-rep} [Y_{Check-NV} Y_{Check-CPMGR} Y_{Check-DPMGR} Y_{Check-APMGR} Y_{Process-logging2} > M_{EOP} \quad (6)$$

첫 번째 도달 가능한 상태의 (3) 의

$$M_{Begin-rep} [Y_{Check-CV} Y_{Process-logging1} > M_{EOP}$$

은 다음 (7)과같이 표현 할 수 있다.

$$M_{Begin-rep} [Y_{Check-CV} > M_{After-failCV} [Y_{Process-logging} > M_{EOP} \quad (7)$$

식(7)에서  $M_{Begin-rep}[Y_{Chek-CV} > M_{After-failCV}]$  이  $W(M_{Begin-rep}) = W(M_{After-failCV})$  임을 증명한다. 그리고

$$\begin{aligned} & \sum_{p \in P} W_p(E(p,t) < request >) \\ &= \sum_{p \in P} W_p(E(t,p) < request1 >) \end{aligned}$$

임을 증명하면 복제시스템의 안정성을 검증하게 된다.

[정의 2]의 [증명]방법에 따라 다음 식을 얻을 수 있다.

$$\begin{aligned} \forall p \in P : M_2(After-failCV) \\ &= (M_1(Begin-rep) - \sum_{(t,b) \in Y} E(p,t) < request1 >) \\ &+ \sum_{(t,b) \in Y} E(t,p) < request1 >. \end{aligned} \quad (8)$$

첫 번째 합은 제거된 토큰들을 나타내고, 두 번째 합은 추가된 토큰들을 나타낸다.  $Y$ 단계의 발생에 의해서  $M_2$ 는  $M_1$ 에서 직접 도달가능하며, (8)식에 의해서 [정의 2]의 (정리 1)을 적용하면 다음 식 (9)와 같다.

$$\begin{aligned} & \sum_{p \in P} W_p(M_2(After-failCV)) \\ &+ \sum_{(t,b) \in Y} E(p,t) < request > \\ &= \sum_{p \in P} W_p(M_1(Begin-rep)) \\ &+ \sum_{(t,b) \in Y} E(t,p) < request > \end{aligned} \quad (9)$$

식 (9)은 가중치함수(weight function)의 선형성(linearity)으로부터 다음 식 (10)을 얻을 수 있다:

$$\begin{aligned} & \sum_{p \in P} W_p(M_2(After-checkCV)) \\ &+ \sum_{p \in P} \sum_{(t,b) \in Y} W_p(E(p,t) < request1 >) \\ &= \sum_{p \in P} W_p(M_1(Begin-rep)) \\ &+ \sum_{p \in P} \sum_{(t,b) \in Y} W_p(E(t,p) < Chek-CV >) \end{aligned} \quad (10)$$

식 (10)은 흐름 성질(flow property)로부터 다음 식 (11)을 얻을 수 있다.

$$\forall (t,b) \in BE : \quad (11)$$

$$\begin{aligned} & \sum_{p \in P} W_p(E(p,t) < request >) \\ &= \sum_{p \in P} W_p(E(t,p) < request >) \end{aligned} \quad (11)$$

식 (11)은 다음 식 (12)과 같이 적용된다.

$$\begin{aligned} & \sum_{(t,b) \in Y} \sum_{p \in P} W_p(E(p,t) < request >) \\ &= \sum_{(t,b) \in Y} \sum_{p \in P} W_p(E(t,p) < request >) \end{aligned} \quad (12)$$

식 (12)은 다음 식 (13)과 같이 다시 작성할 수 있다:

$$\begin{aligned} & \sum_{p \in P} \sum_{(t,b) \in Y} W_p(E(p,t) < request >) \\ &= \sum_{p \in P} \sum_{(t,b) \in Y} W_p(E(t,p) < request >) \end{aligned} \quad (13)$$

위의 식 (13)에서 두 개의  $\sum$ 는 위의 식과 동일하기 때문에 다음 식 (14)과 같은 결론을 얻을 수 있다.

$$\begin{aligned} & \sum_{p \in P} W_p(M_2(After-failCV)) \\ &= \sum_{p \in P} W_p(M_1(Begin-rep)) \end{aligned}$$

즉,

$$W(M_{After-failCV}) = W(M_{Begin-rep}) \quad (14)$$

이다.

다음에  $M_{EOP} \in [M_{Begin-rep} >]$ 을 도달 가능한 마킹(marking)이라고 하고,  $\sigma$ 를  $M_{Begin-rep}$ 에서 시작해서  $M_{EOP}$ 으로 끝나는 발생순서라고 하자.

위의 결과를  $\sigma$ 의

$$M_{Begin-rep}[Y_{Chek-CV} Y_{Process-logging} > M_{EOP}]$$

의 각 단계에 적용하면  $W(M_{EOP}) = W(M_{Begin-rep})$ 이라는 결론을 얻을 수 있다. 그러므로 [정의 2]의 (정리 1)이 증명된다.

이번에는  $\sum_{p \in P} W_p(E(p,t) < b >) = \sum_{p \in P} W_p(E(t,p) < b >)$ 을 증명하기 위해서  $W$ 의 상태 불변식을 결정하고 복제시스템 모니터링 기능은 CPN이 동작하지 않는 바인딩 구성 요소는 가지고 있지 않다고 하자. 이것은 각 바인딩 구성 요소  $(t,b)$ 가 적어도 도달 가능한 하나의  $M$ 을 가지고 있다는 것을 의미한다.

백업 시스템의 안정성 검증을 위한 CPN의 각 도달 가능한 상태들은 다음과 같이 3가지 도달 가능한 상태

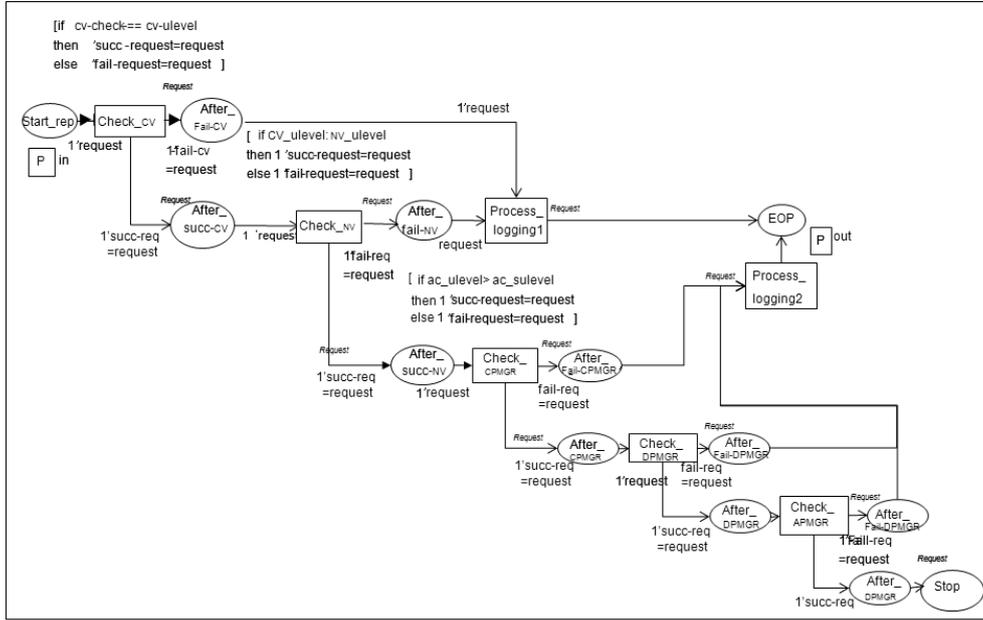


그림 5. 백업 시스템의 CPN 도식화  
Fig. 5. A CPN Diagram of Backup System.

로 표현 할 수 있으며, 각각의 경우는 상태 및 전이 불변 식에 따라 같은 방법으로 그 안정성이 증명된다.

$$M_{Begin-rep} [Y_{Check-CV} Y_{Process-logging1} > M_{EOP}] \quad (15)$$

$$M_{Begin-rep} [Y_{Check-NV} Y_{Check-CPMGR} Y_{Process-logging2} > M_{EOP}] \quad (16)$$

$$M_{Begin-rep} [Y_{Check-NV} Y_{Check-CPMGR} Y_{Check-DPMGR} Y_{Process-logging2} > M_{EOP}] \quad (17)$$

$$M_{Begin-rep} [Y_{Check-NV} Y_{Check-CPMGR} Y_{Check-DPMGR} Y_{Check-APMGR} Y_{Process-logging2} > M_{EOP}] \quad (18)$$

첫 번째 도달 가능한 상태 (15)의

$$M_{Begin-rep} [Y_{Check-CV} Y_{Process-logging1} > M_{EOP}]$$

은 다음 (19)과 같이 표현 할 수 있다.

$$M_{Begin-rep} [Y_{Check-CV} > M_{After-failCV} [Y_{Process-logging} > M_{EOP}]] \quad (19)$$

식 (19)에서  $M_{Begin-rep} [Y_{Check-CV} > M_{After-failCV}]$ 의  $W(M_{Begin-rep}) = W(M_{After-failCV})$ 임을 증명한다. 그리고

$$\sum_{p \in P} W_p(E(p,t) < request >) = \sum_{p \in P} W_p(E(t,p) < request1 >)$$

임을 증명하면 복제시스템의 안정성을 검증하게 된다. [정의 2]의 [증명] 방법에 따라 식 (20)을 얻을 수 있다.

$$\begin{aligned} & \forall p \in P : M_2(After-fail CV) \\ &= (M_1(Begin-rep) - \sum_{(t,b) \in Y} E(p,t) < request1 >) \\ &+ \sum_{(t,b) \in Y} E(t,p) < request1 >. \end{aligned} \quad (20)$$

첫 번째 합은 제거된 토큰들을 나타내고, 두 번째 합은 추가된 토큰들을 나타낸다. Y단계의 발생에 의해서  $M_2$ 는  $M_1$ 에서 직접 도달가능하며, (20)식에 의해서 [정의 2]의 (정리 1)을 적용하면 다음 식 (21)과 같다.

$$\begin{aligned} & \sum_{p \in P} W_p(M_2(After-fail CV)) \\ &+ \sum_{(t,b) \in Y} E(p,t) < request > \\ &= \sum_{p \in P} W_p(M_1(Begin-rep) + \sum_{(t,b) \in Y} E(t,p) < request >) \end{aligned} \quad (21)$$

식 (21)은 가중치 함수(weight function)의 선형성(linearity)으로부터 식 (22)을 얻을 수 있다:

$$\begin{aligned} & \sum_{p \in P} W_p(M_2(\text{After} - \text{chek}CV)) \\ & + \sum_{p \in P} \sum_{(t,b) \in Y} W_p(E(p,t) < \text{request1} >) \\ & = \sum_{p \in P} W_p(M_1(\text{Begin} - \text{rep})) \\ & + \sum_{p \in P} \sum_{(t,b) \in Y} W_p(E(t,p) < \text{Chek} - CV >) \end{aligned} \quad (22)$$

식 (22)은 흐름 성질(flow property)로부터 식 (23)을 얻을 수 있다.

$$\begin{aligned} & \forall (t,b) \in BE \quad : \\ & \sum_{p \in P} W_p(E(p,t) < \text{request} >) \\ & = \sum_{p \in P} W_p(E(t,p) < \text{request} >) \end{aligned} \quad (23)$$

식 (23)은 식 (24)과 같이 적용된다.

$$\begin{aligned} & \sum_{(t,b) \in Y} \sum_{p \in P} W_p(E(p,t) < \text{request} >) \\ & = \sum_{(t,b) \in Y} \sum_{p \in P} W_p(E(t,p) < \text{request} >) \end{aligned} \quad (24)$$

식 (24)은 식 (25)과 같이 다시 작성할 수 있다:

$$\begin{aligned} & \sum_{p \in P} \sum_{(t,b) \in Y} W_p(E(p,t) < \text{request} >) \\ & = \sum_{p \in P} \sum_{(t,b) \in Y} W_p(E(t,p) < \text{request} >) \end{aligned} \quad (25)$$

위의 식 (25)에서 두 개의  $\sum$ 는 위의 식과 동일하기 때문에 식 (26)과 같은 결론을 얻을 수 있다.

$$\begin{aligned} & \sum_{p \in P} W_p(M_2(\text{After} - \text{fail}CV)) \\ & = \sum_{p \in P} W_p(M_1(\text{Begin} - \text{rep})) \end{aligned}$$

즉,

$$W(M_{\text{After} - \text{fail}CV}) = W(M_{\text{Begin} - \text{rep}}) \quad (26)$$

다음에  $M_{EOP} \in [M_{\text{Begin} - \text{rep}} >]$ 을 도달 가능한 마킹(marking)이라고 하고,  $\sigma$ 를  $M_{\text{Begin} - \text{rep}}$ 에서 시작해서  $M_{EOP}$ 으로 끝나는 발생순서라고 하자.

위의 결과를  $\sigma$ 의

$$M_{\text{Begin} - \text{rep}}[Y_{\text{Chek} - CV} Y_{\text{Process} - \text{logging}} > M_{EOP}$$

의 각 단계에 적용하면  $W(M_{EOP}) = W(M_{\text{Begin} - \text{rep}})$ 이라는 결론을 얻을 수 있다. 그러므로 [정의 2]의 (정리

1)이 증명된다.

$$\text{이번에는 } \sum_{p \in P} W_p(E(p,t) < b >) = \sum_{p \in P} W_p(E(t,p) < b >)$$

를 증명하기 위해서  $w$ 의 상태 불변식을 결정하고 복제시스템 모니터링 기능은 CPN이 동작하지 않는 바인딩 구성 요소는 가지고 있지 않다고 하자. 이것은 각 바인딩 구성 요소( $t,b$ )가 적어도 도달 가능한 하나의  $M_1$ 을 가지고 있다는 것을 의미한다.

$M_{\text{After} - \text{fail}CV}$ 을  $M_{\text{Begin} - \text{rep}}[t,b > M_{\text{After} - \text{fail}CV}$ 에 의해 결정되는 마킹이라고 하자. 여기서  $t$ 와  $b$ 를 다시 정의하면,  $t$ 는 전이(transition)의 구성 요소(element)이다.

[그림 5]에서 첫 번째 발생 가능한 단계  $Y_{\text{Chek} - CV}$ 의 가드 식  $G(t)$ 는  $G(t) = (\text{source} - \text{system} = \text{target} - \text{system})$ 이어야 하며, 노드 값에 따라서 바인딩 원소  $\langle b \rangle$ 값이 succ-request 또는 fail-request로 결정된다. 단 succ-request와 fail-request 그리고 request는 동일한 request의 컬러 타입을 갖는다.

그러므로  $M_{\text{Begin} - \text{rep}}[t,b > M_{\text{After} - \text{fail}CV}$ 에서  $t$ 의 조건식에서 근원지 시스템과 대상 시스템의 임계값이 같이 않다면  $b$ 는 fail-request로서  $M_{\text{After} - \text{fail}CV}$ 상태의 마킹값을 갖게 된다. 그러므로  $M_{\text{After} - \text{fail}CV}$ 는  $M_{\text{Begin} - \text{rep}}[t,b > M_{\text{After} - \text{fail}CV}$ 에 의해 결정되는 마킹이므로, 앞서 증명된 결과에 따라서

$$W(M_{\text{After} - \text{fail}CV}) = W(M_{\text{Begin} - \text{repr}})$$

이 [정의 2]의 (정리 2)에 따라 다음 식(27)이 적용되어, (28)와 같이 증명된다.

$$\forall M \in [M_0 > : \sum_{p \in P} W_p(M(p)) = \sum_{p \in P} W_p(M_0(p)). \quad (27)$$

$$\sum_{p \in P} W_p(E(p,t) < \text{request} >) = \sum_{p \in P} W_p(E(t,p) < \text{request} >) \quad (28)$$

백업시스템의 각 도달 가능한 상태들은 다시 세부적으로 다음과 같이 각각 정의 할 수 있다.

$$M_{\text{Begin} - \text{rep}}[Y_{\text{Chek} - CV} Y_{\text{Process} - \text{logging}1} > M_{EOP} \quad (29)$$

$$M_{\text{Begin} - \text{rep}}[Y_{\text{Chek} - NV} Y_{\text{Chek} - \text{CPMGR}} Y_{\text{Process} - \text{logging}2} > M_{EOP} \quad (30)$$

$$M_{Begin-rep} [Y_{Check-NV} Y_{Check-CPMGR} Y_{Check-DPMGR} Y_{Process-logging2} > M_{EOP}] \quad (31)$$

$$M_{Begin-rep} [Y_{Check-NV} Y_{Check-CPMGR} Y_{Check-DPMGR} Y_{Check-APMGR} Y_{Process-logging2} > M_{EOP}] \quad (32)$$

첫 번째 도달 가능한 상태의 (29)

$M_{Begin-rep} [Y_{Check-CV} Y_{Process-logging1} > M_{EOP}]$ 은 다음 (33)과 같이 표현 할 수 있다.

$$M_{Begin-rep} [Y_{Check-CV} > M_{After-failCV} Y_{Process-logging} > M_{EOP}] \quad (33)$$

식 (33)에서  $M_{Begin-rep} [Y_{Check-CV} > M_{After-failCV}]$ 의  $W(M_{Begin-rep}) = W(M_{After-failCV})$  임을 증명한다. 그리고

$$\begin{aligned} & \sum_{p \in P} W_p(E(p,t) < request >) \\ &= \sum_{p \in P} W_p(E(t,p) < request1 >) \end{aligned}$$

임을 증명하면 백업시스템의 안정성을 검증하게 된다.

같은 방법으로 [정의 2]의 [증명]방법에 따라 다음 식 (34)을 얻을 수 있다.

$$\begin{aligned} & \forall p \in P : M_2(After-failCV) \\ &= (M_1(Begin-rep) - \sum_{(t,b) \in Y} E(p,t) < request1 >) \\ &+ \sum_{(t,b) \in Y} E(t,p) < request1 >. \end{aligned} \quad (34)$$

첫 번째 합은 제거된 토큰들을 나타내고, 두 번째 합은 추가된 토큰들을 나타낸다.  $Y$ 단계의 발생에 의해서  $M_2$ 는  $M_1$ 에서 직접 도달가능하며, (34)식에 의해서 [정의 2]의 (정리 1)를 적용하면 다음 식 (35)과 같다:

$$\begin{aligned} & \sum_{p \in P} W_p(M_2(After-failCV)) \\ &+ \sum_{(t,b) \in Y} E(p,t) < request > \\ &= \sum_{p \in P} W_p(M_1(Begin-rep)) \\ &+ \sum_{(t,b) \in Y} E(t,p) < request > \end{aligned} \quad (35)$$

식 (35)은 가중치 함수(weight function)의 선형성(linearity)으로부터 다음 식 (36)을 얻을 수 있다.

$$\begin{aligned} & \sum_{p \in P} W_p(M_2(After-failCV)) \\ &+ \sum_{p \in P(t,b) \in Y} W_p(E(p,t) < request1 >) \\ &= \sum_{p \in P} W_p(M_1(Begin-rep)) \\ &+ \sum_{p \in P(t,b) \in Y} W_p(E(t,p) < Check-CV >) \end{aligned} \quad (36)$$

식 (36)은 흐름 성질(flow property)로부터 식 (37)을 얻을 수 있다.

$$\begin{aligned} & \forall (t,b) \in BE : \\ & \sum_{p \in P} W_p(E(p,t) < request >) \\ &= \sum_{p \in P} W_p(E(t,p) < request >) \end{aligned} \quad (37)$$

식 (37)은 식 (38)과 같이 적용된다.

$$\begin{aligned} & \sum_{(t,b) \in Y_{p \in P}} W_p(E(p,t) < request >) \\ &= \sum_{(t,b) \in Y_{p \in P}} W_p(E(t,p) < request >) \end{aligned} \quad (38)$$

식 (38)은 식 (39)과 같이 다시 작성할 수 있다.

$$\begin{aligned} & \sum_{p \in P(t,b) \in Y} W_p(E(p,t) < request >) \\ &= \sum_{p \in P(t,b) \in Y} W_p(E(t,p) < request >) \end{aligned} \quad (39)$$

위의 식 (39)에서 두 개의  $\sum$ 는 위의 식과 동일하기 때문에 다음 식 (40)과 같은 결론을 얻을 수 있다.

$$\begin{aligned} & \sum_{p \in P} W_p(M_2(After-failCV)) \\ &= \sum_{p \in P} W_p(M_1(Begin-rep)) \end{aligned}$$

즉,

$$W(M_{After-failCV}) = W(M_{Begin-rep}) \quad (40)$$

다음에  $M_{EOP} \in [M_{Begin-rep}]$ 을 도달 가능한 마킹(marking)이라고 하고,  $\sigma$ 를  $M_{Begin-rep}$ 에서 시작해서  $M_{EOP}$ 으로 끝나는 발생순서라고 하자.

위의 결과를  $\sigma$ 의

$$M_{Begin-rep} [Y_{Check-CV} Y_{Process-logging} > M_{EOP}]$$

의 각 단계에 적용하면  $W(M_{EOP}) = W(M_{Begin-rep})$ 이라는 결론을 얻을 수 있다. 그러므로 [정의 2]의 (정리

1)이 증명된다.

$$\text{이번에는 } \sum_{p \in P} W_p(E(p,t) < b >) = \sum_{p \in P} W_p(E(t,p) < b >)$$

을 증명하기 위해서  $W$ 의 상태 불변 식을 결정하고 백업 시스템기능은 CPN이 동작하지 않는 바인딩 구성 요소는 가지고 있지 않다고 하자. 이것은 각 바인딩 구성 요소( $t, b$ )가 적어도 도달 가능한 하나의  $M_1$ 을 가지고 있다는 것을 의미한다.

$M_{After-failCV}$ 을  $M_{Begin-rep}[t, b > M_{After-failCV}$ 에 의해 결정되는 마킹이라고 하자.

여기서  $t$ 와  $b$ 를 다시 정의하면.  $t$ 는 전이(transition)의 구성 요소(element)이다.

즉, [그림 5]에서 첫 번째 발생 가능한 단계  $Y_{Check-CV}$ 의 가드 식  $G(t)$ 는  $G(t)=(source-system == target-system)$ 이어야 하며, 노드 값에 따라서 바인딩 원소 $<b>$ 값이 succ-request 또는 fail-request로 결정된다. 단 succ-request와 fail-request 그리고 request는 동일 한 request의 컬러 타입을 갖는다.  $M_{Begin-rep}[t, b > M_{After-failCV}$ 에서  $t$ 의 조건식에서 근원지 시스템과 대상시스템의 임계값이 같이 않다면  $b$ 는 fail-request로서  $M_{After-failCV}$ 상태의 마킹 값을 갖는다.

그러므로  $M_{After-failCV}$ 는  $M_{Begin-rep}[t, b > M_{After-failCV}$ 에 의해 결정되는 마킹이므로, 앞서 증명된 결과에 따라 식  $W(M_{After-failCV}) = W(M_{Begin-rep})$ 는 [정의 2]의 (정리 2)에 따라 식(41)이 적용되어, (42)와 같이 증명된다.

$$\forall M \in [M_0 > : \sum_{p \in P} W_p(M(p)) = \sum_{p \in P} W_p(M_0(p)). \quad (41)$$

$$\sum_{p \in P} W_p(E(p,t) < request >) = \sum_{p \in P} W_p(E(t,p) < request >) \quad (42)$$

복제시스템 모니터링 기능의 각 도달 가능한 상태들은 다시 세부적으로 다음과 같이 정의 할 수 있다.

$$\begin{aligned} M_{Begin-rep}[Y_{Check-CV}Y_{Process-logging1} > M_{EOP} \\ M_{Begin-rep}[Y_{Check-CV} > M_{failCV} \\ M_{failCV}[Y_{Process-logging1} > M_{EOP} \end{aligned} \quad (43)$$

$$\begin{aligned} M_{Begin-rep}[Y_{Check-CV}Y_{Check-NV}Y_{Process-logging2} > M_{EOP} \\ M_{Begin-rep}[Y_{Check-CV} > M_{After-sucCV} \end{aligned}$$

$$\begin{aligned} M_{After-sucCV}[Y_{Check-madNV} > M_{After-failNV} \\ M_{After-failNV}[Y_{Process-logging} > M_{EOP} \end{aligned} \quad (44)$$

$$\begin{aligned} M_{Begin-rep}[Y_{Check-CV}Y_{Check-NV}Y_{Process-logging2} > M_{After-CPMGR} \\ M_{Begin-rep}[Y_{Check-CV} > M_{After-sucCV} \\ M_{After-sucCV}[Y_{Check-NV} > M_{After-sucNV} \\ M_{After-sucNV}[Y_{Process-logging2} > M_{After-CPMGR} \end{aligned} \quad (45)$$

$$\begin{aligned} M_{Begin-rep}[Y_{Check-CV}Y_{Check-NV}Y_{Process-logging2} > M_{After-DPMGR} \\ M_{Begin-rep}[Y_{Check-CV} > M_{After-sucCV} \\ M_{After-sucCV}[Y_{Check-NV} > M_{After-sucNV} \\ M_{After-sucNV}[Y_{Process-logging2} > M_{After-DPMGR} \end{aligned} \quad (46)$$

$$\begin{aligned} M_{Begin-rep}[Y_{Check-CV}Y_{Check-NV}Y_{Process-logging2} > M_{After-APMGR} \\ M_{Begin-rep}[Y_{Check-CV} > M_{After-sucCV} \\ M_{After-sucCV}[Y_{Check-NV} > M_{After-sucNV} \\ M_{After-sucNV}[Y_{Process-logging2} > M_{After-APMGR} \end{aligned} \quad (47)$$

복제 시스템 모니터링 기능의 초기 상태에서 발생 가능한 상태(place)들은 상태 불변 식에 의해 같은 방법으로 안정성이 증명된다. 또한 CPN의 속성 중에서 보존성은 전이의 입력 수는 출력 수와 같아야 하는데,  $|I(t_i)| = |O(t_i)|$ 이므로 복제 시스템은 전이(transition)의 조건식  $G(t)$ 에 따라 입력 토큰의 개수와 출력 토큰의 개수가 같으므로 제안하는 백업시스템의 안정성이 증명된다.

## V. 결론 및 향후 연구 방향

본 연구에서는 재해에 대비한 고가용성(high availability) 백업시스템을 설계하였으며, 시스템의 동작과정은 온라인 개장 기록(redo log)을 처리기록(transaction log) 데이터베이스로 실시간 백업함으로써 기존 백업방법에서 누락 될 수 있는 자료보관소 기록(archive log)으로 백업함으로써, 데이터의 실시간 복구 가 가능하고 복구 시 온라인으로 개장 기록(redo log)까지 복구가 가능하여 데이터의 손실을 최소화 할 수 있다. 설계된 고성능 백업시스템의 안정성을 검증하기 위해서 CPN(Coloured Petri Net)의 상태 불변식을 도입하

여 백업시스템의 각 단계는 CPN의 그래픽으로 도식화하고, 각 단계를 CPN의 정의와 수식으로 안정성을 검증하였다.

### 참 고 문 헌

- [1] 이문구, “침입 차단 방화벽 시스템을 위한 FTP 프로토콜 모델의 설계 및 안정성 검증” 송실 대학교 대학원 박사학위 논문, 1999. 12.
- [2] 박성원, 이문구, 이남용, “클라우드 컴퓨팅 기반의 고가용성 복제시스템의 구현”, 대한전자공학회 논문지, 제48 권 IE 편 제4호. 2011.
- [3] 이문구, “CPN 기반의 침입방지시스템 보안모델의 안정성 검증” 대한전자공학회 논문지, 제48 권 CI 편, 제3호. 2011.
- [4] 박성원, “클라우드 컴퓨팅 기반의 고가용성 복제 및 백업 시스템에 관한 실증적 연구”, 송실대학교 대학원 박사학위 논문, 2012. 6.
- [5] 김경백, “높은 데이터 유효성을 보장하기 위한 행동-인지 피어-투-피어 프로토콜 : Behavior-aware P2P Protocol for High Data Availability” Korea Advanced Institute of Science and Technology, 2007.
- [6] Kurt Jensen, “Colored Petri Nets. Basic Concepts, Analysis Methods and Practical Use”. Volume 1, 2, 3: Basic Concepts, EATCS monographs on Theoretical Computer Science, Springer-Verlag 1992.
- [7] Vijay Gehlot, “Systems Modeling and Analysis Using Colored Petri Nets”, IEEE Systems Conference 2008.
- [8] Aggarwal, Nidhi, “Achieving high availability with commodity hardware and software”, The University of Wisconsin - Madison, 2008
- [9] [http://www.daimi.au.dk/CPnets/intro/example\\_indu.html](http://www.daimi.au.dk/CPnets/intro/example_indu.html)
- [10] Ji, Minwen, “Availability, scalability and a cost-effectiveness of cluster-based Internet infrastructures”, Princeton University, 2001.

### 저 자 소 개



이 문 구(평생회원)  
1984년 송실대학교  
전자계산학 (학사)  
1993년 이화여자대학교 대학원  
전산교육학 (석사)  
2000년 송실대학교 대학원  
컴퓨터시스템 (공학 박사)

2000년 3월~현재 김포대학 IT 학부  
인터넷정보과 부교수

<주관심분야 : 알고리즘, 인터넷 보안, 전자상거래 보안, 시스템 보안, 네트워크 보안>