시스템 오류 발생률 분석

성순용*

An Analysis of System Error Rate

Soonyong Seong*

이 논문은 2006학년도 부산외국어대학교 학술연구 조성비에 의하여 연구되었음

요 약

교착상태의 발생 주기 및 확률은 교착상태를 다루는 알고리즘 설계 시 많은 영향을 미친다. 그러나 프로세스나 자원의 성격, 자원 요구나 반환 연산 방식, 프로세스 개수 등의 성질이 교착상태 발생에 어떻게 영향을 미치는지 분석하는 게 쉽지 않아 이 분야에 대한 연구가 매우 부족하다. 이 논문은 자원 할당 상태를 (a,b)t로 표현하는 상태 모델을 이용하여 상태의 개수를 획기적으로 감소시켰다. 또한 시스템 분석에 있어서 자원의 오류 발생 비율과 복구 비율이 미치는 영향도 함께 포함할 수 있도록 설계하였다. 그 결과 교착상태의 평균 발생 주기, 요구연산이 보류되거나 교착상태를 유발할 확률, 사이클의 길이가 2인 교착상태가 발생할 확률 등과 같은 각종 수식을 구하였다.

ABSTRACT

The frequency and probability of deadlock are influential factors in the design of algorithms for deadlock. However, little work has been done in this area because it's not easy to analyze how factors such as the characteristics of process or resource, resource request and release patterns, or the number of process affect the occurrence of deadlock. This study was designed to reduce remarkably the number of state by adapting the model 'state (a,b)t' to represent the resource allocation state, as well as to include the effect of resource error rate and recovery rate in the system analysis. Various formulas about deadlock occurrence were resulted in this study such as the average time interval of deadlock, the probability that a process requesting a resource waits or deadlocks, and the probability that a request deadlocks in a cycle of length 2.

키워드

error rate, deadlock, state transition, resource allocation graph

* 부산외국어대학교 컴퓨터공학과

접수일자 2008. 12. 26

Ⅰ. 서 론

컴퓨터 시스템의 성능을 저하시키는 교착상태는 가능한 빨리 발견하고 제거시켜야 하나, 발생 여부를 점검하는데 따르는 부하 증가도 시스템 성능을 떨어지게 한다. 따라서 교착상태를 가능한 빨리 발견해 내면서도 시스템 부하를 줄일 수 있는 최적의 점검 주기를 선택하는 문제가 제기된다.

적절한 주기를 선택하기 위해서는 시스템의 제반 여건들, 즉 프로세스나 자원의 수, 각 프로세스의 자원 이용 특성 등에 따른 교착상태 발생 빈도의 변화를 비교 분석한 결과가 필요하다. 이 결과를 확률적 모델을 이용하여 계산하기 위해 많은 연구가 진행되어 왔다[1,2]. 특히 서로 다른 자원 할당 상태의 수를 현저하게 줄일 수 있는 새로운 자원 할당 표현 모델이 제시되어, 이를 이용함으로써 이전에는 거의 불가능했던 계산들을 가능케 하고 있다[3,4]. 즉 시스템의 자원 할당 상태를 자원할당그래프로 표현했을 때 요구간선과 할당간선의 수가 각각 a개, b 개이고 교착상태가 아니면 이를 상태 (a,b)로 표시하는 모델이 그것이다. 이 표현 기법은 어느 프로세스와 어느 자원 사이에 간선이 연결되어 있는지를 구체적으로 명시하지 않고, 단지 요구나 할당의 정도에 따라 시스템 상태를 분류하고 있다.

앞서 이 모델을 이용하여 시스템이 안정 상태(steady state)에 도달하였을 때 보이는 특성들 - 교착상태의 발생 확률과 발생 주기, 시스템의 평균 상태 (즉 보류 프로세스 비율, 자원 이용률 등) 및 기타 여러 특성들을 분석한 바 있다[5].

이전 논문에서는 자원의 요구 연산과 반환 연산의 발생 비율만을 고려하였으나, 본 논문에서는 자원의 오류 발생도 표현할 수 있는 모델로 확장하였다. 새로 제시된 모델에서는 자원의 오류 발생비율과 복구비율이 미치는 특성을 추가 반영하여 분석하고자 한다.

Ⅱ. 시스템 정의

앞으로 분석할 시스템을 다음과 같이 정의한다.

시스템에는 n 개의 동종 프로세스와 서로 다른 r 종류의 재사용 가능 자원이 한 단위씩 존재한다. 시스템 전체에서 프로세스들의 자원 요구 및 반환연산이 반복 수행되는데, 이때 연산을 수행하는 프로세스는 자신이 현재 가지고 있지 않은 자원들을 한 번에 하나씩 요구하며, 자신이 갖고 있는 자원들을 한 번에 하나씩 반환한다.

요구, 또는 반환연산이 이루어질 때, 해당 연산이 가능한 프로세스들 모두에게 같은 확률로 기회가 주어지고, 각 프로세스의 요구연산이나 반환연산에서 자원의 선택도 같은 확률로 주어진다. 시스템에서의 자원에 대한 요구 및 반환연산은 그 연산이 가능한 프로세스가 하나 이상 존재할 때 그 평균 수행비율이 각각 λ, μ인 Poisson 분포의 빈도로 발생한다.

자원의 요구연산 시 해당 자원이 다른 프로세스에 이미 할당된 상태이거나 오류상태이면 대기하게 되는데, 이 상태를 보류 프로세스라 하고 보류 프로세스가 아닌 프로세스를 실행 프로세스라 한다. 실행 프로세스만이 새로운 자원을 하나 더 요구하거나 자신이 보유한 자원 중의 하나를 반환할 수 있다.

자원에는 오류가 발생할 수 있으며, 오류가 발생하면 이를 복구하기 전에는 프로세스에 할당할 수 없다. 시스템 전체에서 자원의 오류 발생 및 복구가 반복 수행되며, 이때 오류는 한 번에 하나씩 발생하고, 복구도 한 번에 하나씩 이루어진다. 오류의 발생이나 복구 시, 자원 모두에게 같은 확률로 주어진다. 시스템에서의 자원에 대한 오류 발생 및 복구 작업은 현재 해당 작업이 가능한 자원이 존재할 때 그 평균 발생비율이 각각 σ, τ인 Poisson 분포의 빈도로 발생한다.

오류가 발생하는 자원 시스템을 표기하기 위하여 시스템 상태를 $(a,b)_t$로 나타낸다. t는 현재 오류상태에 있는 자원의 수를 표시한다. 시스템의 자원 할당 상태는 즉시할당상태로 가정하여, 요구한 자원이 현재 할당된 상태나 오류상태가 아니면 바로 할당받는다. 오류로부터 복구된 자원도 요구하는 프로세스가 존재하면 즉시 할당한다. 그 결과 요구간선의 수는 보류 프로세스의 수가 되어 상태 $(a,b)_t$에서 a는 보류 프로세스의 수, b는 할당된 자원의 수를 표시하게 된다. 즉 상태 $(a,b)_t$에서는 시스템에서 발생하는 요구연산 또는 반환연산이 n-a 개의 상호 독립적인 실행 프로세스들 중의 하나에서 발생하는 것이다.

교착상태는 발생 즉시 발견되며, 이에 포함된 보류 프로세스들 중에서 임의로 한 프로세스를 종료시킴으로써 교착상태를 제거한다고 가정한다. 이미 할당되어 있는 자원에 오류가 생기는 경우에도 해당 자원을 보유한

프로세스를 종료시킨다. 종료된 프로세스는 아무 자원도 보유하지 않은 상태에서 다시 시작한다. 또한 교착상태의 발생 확률 및 주기에 중점을 두기 위해 교착상태 및 오류의 발견 및 회복, 자원의 할당 및 반환에 소요되는 시간은 무시할 수 있다고 가정한다.

Ⅲ. 안정 상태에서의 상태 $(a,b)_t$의 확률

1) 상태의 총 개수

시스템에서 가능한 서로 다른 상태 $(a,b)_t$는 총 $n(r+1)+(n+1)r(r+1)/2$ 개이다.

$$0 \le a < n,\ 0 \le b \le r,\ t=0$$
$$0 \le a \le n,\ 0 \le b < r,\ 1 \le t \le r-b$$

오류 자원은 할당 불가능하여 $t \le r-b$가 된다.

2) 상태 간의 전이 비율

상태 $(a,b)_t$에서 특정 실행 프로세스가 자원을 i 개 보유하고 있을 확률을 $P_i(a,b)_t$로 표시하기로 한다. 그리고 상태 $(a,b)_t$에서 시스템 전체에서의 요구연산과 반환연산의 발생비율을 각각 $R_{req}(a,b)_t$, $R_{rls}(a,b)_t$로 표기한다.

상태 $(a,b)_t$에서 전이되는 다음 상태는 다음과 같이 결정된다. 반환연산의 경우 해당 자원을 기다리는 프로세스가 있으면 한 프로세스가 보류 상태에서 깨어나게 되며(상태 $(a-1,b)_t$로 전이), 기다리는 프로세스가 없으면 그대로 시스템에 반환된다(상태 $(a,b-1)_t$로 전이). 요구연산의 경우는 비 할당, 비 오류 상태에 있는 자원이면 바로 할당받고(상태 $(a,b+1)_t$로 전이), 다른 프로세스에 할당되어 있는 자원이면서 사이클을 형성하지 않거나 오류 자원이면 보류 상태가 되며(상태 $(a+1,b)_t$로 전이), 사이클이 형성되면 교착상태가 발생하는데 이때 임의로 선택된 종료 프로세스가 반환한 자원들이 α 개의 보류 프로세스들을 깨우고 나머지 β 개가 그대로 반환되면 상태 $(a-\alpha,b-\beta)_t$로 전이하게 된다. 오류가 비 할당 상태의 자원에 발생하면 상태 $(a,b)_{t+1}$로 전이하고, 할당된 상태의 자원에 발생하면 그 자원을 보유한 프로세스가 종료되며 반환하는 나머지 자원들이 α 개의 보류 프로세스들을 깨우고 β개가 그대로 반환되면 상태 $(a-\alpha,b-\beta-1)_{t+1}$로 전이하게 된다. 오류로부터 복구된 자원을 기다리는 프로세스가 있으면 한 프로세스가 보류 상태에서 깨어나게 되며(상태 $(a-1,b+1)_{t-1}$로 전이), 기다리는 프로세스가 없으면 상태 $(a,b)_{t-1}$로 전이한다.

결국 반환연산, 요구연산, 오류 발생 및 복구 각각에 대해 전이되는 다음 상태는 위와 같은 상황이 발생하는 확률에 의해 결정되므로 이 확률을 계산할 수 있어야 한다. 상태 $(a,b)_t$는 여러 상이한 자원할당그래프를 내포하고 있다. 필요한 확률 계산이 가능하도록 하기 위해, 시스템의 자원 할당 상태가 $(a,b)_t$일 때 실제 자원할당그래프가 특정한 상태일 확률은 가능한 모든 자원할당그래프에 대해서 같은 값을 갖는다고 가정한다. 이 가정에 의해 상태 $(a,b)_t$가 임의의 상황을 만족시킬 확률은 상태 $(a,b)_t$가 표현하고 있는 서로 다른 자원할당그래프의 총 경우의 수에서 그 상황을 만족시키는 경우의 수 비율로 계산할 수 있다.

상태 $(a,b)_t$가 표현하고 있는 자원할당그래프의 총 경우의 수는 다음과 같다.

상태 $(a,b)_t$의 자원할당그래프 총 경우의 수

$$= \binom{n}{a}\binom{r}{b}\binom{r-b}{t} X_n(a,b)_t \qquad \text{<식1>}$$

<식1>에서 보류 프로세스 a개, 할당 자원 b개, 그리고 요구 간선만 연결 가능한 오류 자원 t개가 각각 결정된 상태의 경우의 수를 $X_n(a,b)_t$로 표기하였다.

$$X_n(a,b)_t = \begin{cases} \displaystyle\sum_{i=0}^{a}\binom{a}{i} t^i (n-a+i) \bullet \sum_{j=0}^{\min(a-i,\, b-1)} {}_{a-i}P_j\ {}_{b-1}P_j\ X_n(a-i-j,\, b-j-1)_{j+1} & ,\ 1 \le b \le r \\ t^a & ,\ b=0 \end{cases} \qquad \text{<식2>}$$

예를 들어 프로세스와 자원이 각각 10 개, 요구간선과 할당간선이 각각 3 개, 오류 상태의 자원이 2 개인 자원할당그래프의 개수를 구해 보자. <식2>에서 $X_{10}(3,3)_2$ =98,580이고, 이를 <식1>에 대입하면 약 300억 개나 된다. 즉 300억 개에 달하는 상태가 하나의 상태 (3,3)₂로 표

현됨으로써 상태 개수가 획기적으로 감소함을 확인할 수 있다.

다음엔 상태 $(a,b)_t$에서 특정 실행 프로세스가 m 개의 자원을 보유하고 있을 경우의 수를 계산해 보자.

한 실행프로세스가 m개의 자원 보유할 경우의 수

$$= \binom{n}{a}\binom{r}{b}\binom{r-b}{t}\binom{b}{m} X_{n-1}(a, b-m)_{m+t}$$

위에서 구한 값으로부터 $P_m(a,b)_t$를 계산하면

$$P_m(a,b)_t = \frac{\binom{b}{m} X_{n-1}(a,b-m)_{m+t}}{X_n(a,b)_t}$$

이 된다. 비슷한 방법으로 각 상태 간의 전이 확률 및 전이 비율을 구할 수 있다.

① 반환연산

전술한 바와 같이 상태 $(a,b)_t$에서 반환연산의 평균 수행 비율은 $R_{rls}(a,b)_t$이며, 이로 인해 하나의 보류 프로세스가 깨어나거나(*WKE* 전이), 자원을 하나 반환하는(*RLS* 전이) 상태 전이가 유발된다. *WKE* 전이, *RLS* 전이가 발생할 확률을 각각 $P_{WKE}(a,b)_t$, $P_{RLS}(a,b)_t$라 하면 해당 전이 비율은 다음과 같다.

$$\text{상태}(a,b)_t \xrightarrow{WKE} \text{상태}(a-1,b)_t : R_{rls}(a,b)_t \cdot P_{WKE}(a,b)_t$$
$$\text{상태}(a,b)_t \xrightarrow{RLS} \text{상태}(a,b-1)_t : R_{rls}(a,b)_t \cdot P_{RLS}(a,b)_t$$

우선 $R_{rls}(a,b)_t$는 시스템에서의 반환연산이 그 연산이 가능한 프로세스가 하나 이상 존재할 때 평균 수행 비율이 μ인 Poisson 분포를 따른다고 가정한 바에 따라 구할 수 있다. 상태 $(a,b)_t$에서 $n-a$개의 실행 프로세스에게 하나 이상의 자원만 할당되어 있으면 반환 연산이 가능하다. b개의 모든 할당자원이 a개의 보류 프로세스들에게만 할당되는 경우는 교착상태를 제외하면 하나 이상의 자원이 오류 자원을 대기하고 있는 보류 프로세스에게 할당되는 경우이다.

$$R_{rls}(a,b)_t = \left(1 - \frac{X_a(a,b)_t}{X_n(a,b)_t}\right) \cdot \mu \quad , a \neq n,\ b \neq 0$$

$$R_{rls}(a,0)_t = R_{rls}(n,b)_t = 0$$

반환연산에 의해 둘 중의 어느 전이가 발생하는지는 반환하는 자원이 어떤 상황에 있느냐에 따라 결정되므로 전체 경우의 수에서 각각의 상황을 만족시키는 경우의 수를 세면 확률을 계산할 수 있다.

$$P_{RLS}(a,b)_t = \frac{\sum_{i=1}^{b}\sum_{j=0}^{i} \frac{j}{i}\binom{b}{j}\binom{b-j}{i-j} \cdot \sum_{k=i-j}^{a}\binom{a}{k}\left\{{k \atop i-j}\right\}(i-j)!\, X_{n-1}(a-k, b-i)_t}{X_n(a,b)_t - X_{n-1}(a,b)_t} \qquad \text{<식3>}$$

$$P_{WKE}(a,b)_t = 1 - P_{RLS}(a,b)_t$$

<식3>에서 i는 해당 실행 프로세스가 보유한 자원의 수, j는 i 개 중에서 요구간선이 연결되어 있지 않은 자원의 수, k는 i-j 개의 자원들 중의 하나를 기다리는 보류 프로세스의 수를 나타낸다. 이때 $\left\{{k \atop i-j}\right\}$는 i-j 개의 동일 자원들 각각에 대해 k 개의 보류 프로세스로부터의 요구간선이 하나 이상씩 연결되어 있을 경우의 수(Stirling number[6])로서, 자원이 서로 구별되므로 여기에 (i-j)!을 곱해 준다. 분모의 $X_{n-1}(a,b)_t$는 자원을 하나도 갖고 있지 않을 경우로서 이때는 반환연산이 이루어질 수 없으므로 제외된 것이다.

② 요구연산

상태 $(a,b)_t$에서 요구연산의 평균 수행 비율은 $R_{req}(a,b)_t$이며, 이로 인해 하나의 자원이 더 할당되거나(*ALC* 전이), 해당 프로세스가 보류되거나(*BLK* 전이), 교착상태가 발생하게 된다(*DLK* 전이). *ALC*, *BLK*, *DLK* 전이가 발생할 확률을 각각 PALC(a,b)t, PBLK(a,b)t, PDLK(a,b)t라 하면 그 값과 해당 전이 비율은 다음과 같다.

$$\text{상태}(a,b)_t \xrightarrow{ALC} \text{상태}(a,b+1)_t : R_{req}(a,b)_t \cdot P_{ALC}(a,b)_t$$
$$\text{상태}(a,b)_t \xrightarrow{BLK} \text{상태}(a+1,b)_t : R_{req}(a,b)_t \cdot P_{BLK}(a,b)_t$$
$$\text{상태}(a,b)_t \xrightarrow{DLK} \text{교착상태} \quad : R_{req}(a,b)_t \cdot P_{DLK}(a,b)_t$$

우선 $R_{req}(a,b)_t$는 시스템에서의 요구연산이 그 연산이 가능한 프로세스가 하나 이상 존재할 때 평균 수행 비

율이 λ인 Poisson 분포를 따른다고 가정한 바에 따라 구할 수 있다. 실행 프로세스가 하나만 존재하고 이 프로세스가 r개의 모든 자원을 가졌을 때는 요구 연산이 불가능하다.

$$R_{req}(a,b)_t = \lambda \quad a \neq n,\ a = n-1 \text{ and } b \neq r$$
$$R_{req}(n,b)_t = 0$$
$$R_{req}(n-1,r)_0 = \left(1 - r^{n-1}/X_n(n-1,r)_0\right) \cdot \lambda$$
$$P_{ALC}(a,b)_t = \sum_{i=0}^{b} \frac{r-b}{r-i} P_i(a,b)_t \quad b<r \qquad \text{<식4>}$$
$$P_{ALC}(a,r)_t = 0$$
$$P_{DLK}(a,b)_t = \frac{\displaystyle\sum_{i=0}^{b-1} \frac{b-i}{r-i} \binom{b}{i} \sum_{j=1}^{\min(a,b-i)} i\ {}_aP_j\ {}_{b-i-1}P_{j-1} \cdot X_{n-1}(a-j,b-i-j)_{i+j+t}}{\left(X_n(a,b)_t - \delta_{br} r^a\right) \quad \delta_{br} = \begin{cases} 1, & b=r \\ 0, & b \neq r \end{cases}}$$

<식5>

$$P_{BLK}(a,b)_t = 1 - P_{ALC}(a,b)_t - P_{DLK}(a,b)_t$$

<식4>는 해당 실행 프로세스가 i개의 자원을 보유하고 있을 확률에 비 할당된 자원을 요구할 확률을 곱하여 구한 것이다. <식5>는 요구한 자원이 할당된 자원일 확률에 새로운 요구간선의 추가로 인해 사이클이 형성되는 확률을 곱하고 있다. 분모에서 $b=r$일때 r^a를 뺀 것은 모든 자원을 갖고 있는 프로세스에서는 더 이상의 요구 연산이 불가능하기 때문에 그 경우의 수를 뺀 것이다.

DLK 전이에 의해 교착상태가 발생하고 있는데, 교착상태는 발생 즉시 회복되어 다른 상태로 전이된다고 가정한 바 있다. 회복 작업 시 선택된 하나의 종료 프로세스는 강제로 자신의 자원들을 반환하게 된다.

우선 상태 $(a,b)_t$에서 새로운 요구연산으로 인하여 교착상태가 발생하면 보류 프로세스는 총 $a+1$ 개가 된다. 종료 프로세스가 자원을 반환하면 이 중에서 사이클에 포함되어 있던 자원 하나가 *WKE* 전이를 유발하며, 종료 프로세스도 더 이상 보류 프로세스가 아니므로 다음 상태에서의 보류 프로세스 수의 최대값은 $a-1$이다. 그리고 종료 프로세스가 사이클에 포함되어 있지 않은 추가의 자원을 $\alpha+\beta$ 개 더 보유하고 있다면, α 개가 *WKE* 전이를 유발하고 β 개가 *RLS* 전이를 유발하는 경우 다음 상태는 상태 $(a-1-\alpha,b-\beta)_t$가 된다.

$$\underset{(a,b)_t}{\text{상태}} \xrightarrow[DLK\_REC(\alpha,\beta)]{DLK\ \text{교착상태}\ REC(\alpha,\beta)} \underset{(a-1-\alpha,b-\beta)_t}{\text{상태}}$$
$$: R_{req}(a,b)_t \cdot P_{DLK\_REC(\alpha,\beta)}(a,b)_t$$

REC(α,β) 전이는 앞에서 설명한 α, β개의 자원을 갖고 있는 프로세스를 종료시킴으로써 교착상태로부터 회복할 때의 상태 전이를 나타내며, 이때 교착상태처리 시간은 무시할 수 있다고 가정하였으므로, 결국 상태 $(a,b)_t$에서 상태 $(a-1-\alpha,b-\beta)_t$로의 전이 *DLK_REC* (α,β)가 발생할 확률은 아래와 같다.

$$P_{DLK\_REC(\alpha,\beta)}(a,b)_t = P_{DLK}(a,b)_t \cdot P_{REC(\alpha,\beta)}(a+1,b)_t$$
$$P_{REC(\alpha,\beta)}(a,b)_t$$
$$= \frac{\displaystyle\sum_{i=2}^{\min(a,\,b-\alpha-\beta)} \binom{a}{i}\ {}_bP_i \binom{b-i}{\alpha}\binom{b-i-\alpha}{\beta} \cdot \sum_{j=\alpha}^{a-i} \binom{a-i}{j} \alpha! \left\{ {j \atop \alpha} \right\} X_{n-1}(a-i-j,b-\alpha-\beta-i)_{t+i}}{\displaystyle\sum_{i=2}^{\min(a,b)} \binom{a}{i}\ {}_bP_i\ X_n(a-i,b-i)_{t+i}}$$

<식6>

<식6>은 교착상태가 발생한 상태에서 전술한 α, β개의 자원을 가지고 있는 종료 프로세스를 선택할 확률을 나타낸다. 교착상태는 발생 즉시 발견, 회복시키므로 사이클은 하나만 존재하게 되어, 사이클이 하나 존재하는 총 경우의 수에서 원하는 성질을 만족하는 경우의 수를 세어 구한 것이다.

③ 자원 오류 발생과 복구

상태 $(a,b)_t$에서 $t<r$ 이면 오류가 발생 가능하고, $t>0$ 이면 오류 복구가 가능하다. 시스템에서 각각의 발생 비율은 σ와 τ로 정의한 바 있다.
오류가 발생하면 단순히 오류 자원 개수만 증가하거나(*ERR* 전이), 해당 자원을 보유한 프로세스를 종료시키게 된다(*ETM* 전이). 오류로부터 복구된 자원은 하나의 보류 프로세스를 깨우거나(*EWK* 전이), 오류 자원 개수만 줄인다(*ERC* 전이). *ERR*, *ETM*, *EWK*, *ERC* 전이가 발생할 확률을 각각 $P_{ERR}(a,b)_t$, $P_{ETM(\alpha,\beta)}(a,b)_t$, $P_{EWK}(a,b)_t$, $P_{ERC}(a,b)_t$라 하면 그 값과 해당 전이 비율은 다음과 같다.

$$\text{상태}(a,b)_t \xrightarrow{ERR} \text{상태}(a,b)_{t+1} : R_{err}(a,b)_t \cdot P_{ERR}(a,b)_t$$

$$상태(a,b)_t \xrightarrow{ETM(\alpha,\beta)} 상태(a-\alpha,b-\beta-1)_{t+1}$$
$$: R_{err}(a,b)_t \cdot P_{ETM(\alpha,\beta)}(a,b)_t$$
$$상태(a,b)_t \xrightarrow{EWK} 상태(a-1,b+1)_{t-1}$$
$$: R_{erc}(a,b)_t \cdot P_{EWK}(a,b)_t$$
$$상태(a,b)_t \xrightarrow{ERC} 상태(a,b)_{t-1} : R_{erc}(a,b)_t \cdot P_{ERC}(a,b)_t$$
$$R_{err}(a,b)_t = \begin{cases} \sigma & t<r \\ 0 & t=r \end{cases} \quad R_{erc}(a,b)_t = \begin{cases} \tau & t>0 \\ 0 & t=0 \end{cases}$$
$$P_{ERR}(a,b)_t = \frac{r-t-b}{r-t}$$
$$P_{ETM(\alpha,\beta)}(a,b)_t = \frac{b}{r-t} \frac{\binom{b-1}{\alpha}\binom{b-1-\alpha}{\beta}(C_1+C_2)}{X_n(a,b)_t} \quad \text{<식7>}$$
$$C_1 = (n-a)\sum_{i=\alpha}^{a}\binom{a}{i}\left\{ {i \atop \alpha} \right\}\alpha!\, X_{n-1}(a-i,b-\alpha-\beta-1)_{t+1}$$
$$C_2 = a(b-\alpha-\beta-1)\sum_{i=\alpha}^{a-1}\binom{a-1}{i}\left\{ {i \atop \alpha} \right\}\alpha!$$
$$\cdot X_{n-1}(a-i-1,b-\alpha-\beta-1)_{t+1}$$
$$P_{ERC}(a,b)_t = \frac{X_n(a,b)_{t-1}}{X_n(a,b)_t} \quad \text{<식8>}$$
$$P_{EWK}(a,b)_t = 1 - P_{ERC}(a,b)_t$$

<식7>에서 C_1은 오류로 종료되는 프로세스가 실행 프로세스인 경우의 수, C_2는 보류 프로세스인 경우의 수이다. <식8>에서는 복구되는 자원에 요구간선이 하나도 연결되지 않은 수를 센 것이다.

3) 안정 상태에서의 확률 $P(a,b)_t$

지금까지 반환연산, 요구연산, 자원의 오류 발생 및 복구 각각의 경우에 전이되는 다음 상태와 그 전이 비율을 계산하였다.

상태 $(a,b)_t$에서 전이되는 다음 상태는 그 이전 전이 과정과는 무관하게 현재의 상태 $(a,b)_t$에 의해서만 결정되므로 시스템 자원 할당 상태의 변화 과정은 Markov Chain을 형성한다.

또한 상태 $(a,b)_t$에서 다른 상태로 전이하는 비율은 $R_{req}(a,b)_t + R_{rls}(a,b)_t + R_{err}(a,b)_t + R_{erc}(a,b)_t$이고, 다른 상태들로부터 전이되는 비율은 상태 $(a+1,b)_t$에서의 *WKE* 전이비율, 상태 $(a,b+1)_t$에서의 *RLS* 전이비율, 상태 $(a,b-1)_t$에서의 *ALC*전이비율, 상태 $(a-1,b)_t$에서의 *BLK* 전이비율, 그리고 가능한 α, β 조합에 대한 상태 $(a+1+\alpha,b+\beta)_t$에서의 *DLK_REC*(α,β) 전이비율, 상태 $(a,b)_{t-1}$에서의 *ERR* 전이비율, 가능한 α, β 조합 모두에 대해 상태 $(a+\alpha,b+\beta+1)_{t-1}$에서의 *ETM*$(\alpha,\beta)$ 전이비율, 상태 $(a+1,b-1)_{t+1}$에서의 *EWK* 전이비율, 상태 $(a,b)_{t+1}$에서의 *ERC*전이비율 등이 있다. 이와 같은 상태 간의 전이 비율을 이용하면 안정 상태에서 시스템의 자원 할당 상태가 상태 $(a,b)_t$일 확률 $P(a,b)_t$를 계산할 수 있다.

즉 시스템이 안정 상태에 있을 때 상태 $(a,b)_t$에서 다른 상태들로 전이하는 비율 *Flow_rate_out_of*$(a,b)_t$와 다른 상태들로부터 상태 $(a,b)_t$로 전이되는 비율 *Flow_rate_into*$(a,b)_t$는 서로 같아야만 한다[7].

$$\begin{aligned}
&Flow_rate_out_of(a,b)_t \\
&= (R_{req}(a,b)_t + R_{rls}(a,b)_t + R_{err}(a,b)_t + R_{erc}(a,b)_t) \cdot P(a,b)_t \\
&Flow_rate_into(a,b)_t \\
&= R_{rls}(a+1,b)_t\, P_{WKL}(a+1,b)_t\, P(a+1,b)_t \\
&+ R_{rls}(a,b+1)_t\, P_{RLS}(a,b+1)_t\, P(a,b+1)_t \\
&+ R_{req}(a,b-1)_t\, P_{ALC}(a,b-1)_t\, P(a,b-1)_t \\
&+ R_{req}(a-1,b)_t\, P_{BLK}(a-1,b)_t\, P(a-1,b)_t \\
&+ \sum_\alpha\sum_\beta R_{req}(a+1+\alpha,b+\beta)_t \cdot \\
&\quad P_{DLK_REC(\alpha,\beta)}(a+1+\alpha,b+\beta)_t\, P(a+1+\alpha,b+\beta)_t \\
&+ R_{err}(a,b)_{t-1}\, P_{ERR}(a,b)_{t-1}\, P(a,b)_{t-1} \\
&+ \sum_\alpha\sum_\beta R_{err}(a+\alpha,b+\beta+1)_{t-1} \\
&\quad P_{ETM(\alpha,\beta)}(a+\alpha,b+\beta+1)_{t-1}\, P(a+\alpha,b+\beta+1)_{t-1} \\
&+ \\
&R_{erc}(a+1,b-1)_{t+1}\, P_{EWK}(a+1,b-1)_{t+1}\, P(a+1,b-1)_{t+1} \\
&+ R_{erc}(a,b)_{t+1}\, P_{ERC}(a,b)_{t+1}\, P(a,b)_{t+1}
\end{aligned}$$

위의 식으로부터 $P(a,b)_t$가 일단 구해지면 이를 이용하여 다음 절에서와 같은 결과들을 얻을 수 있다.

Ⅳ. 교착상태 분석

시스템이 안정 상태에 있을 때 교착상태로 전이되는 평균 비율 *DL_RATE*는 다음과 같이 구할 수 있다.

$$DL\_RATE = \sum_{all(a,b)_t} P(a,b)_t \cdot R_{req}(a,b)_t \cdot P_{DLK}(a,b)_t$$

<식9>

이로부터 교착상태가 발생하는 평균 간격은

$$DL\_TIME = \frac{1}{DL\_RATE}$$

이 된다.

다음 요구연산이 보류될 확률은 요구연산의 평균 발생비율에서 *BLK* 전이와 *DLK* 전이가 차지하는 비율을 구하면 된다.

$$PROBABILITY\_A\_REQUEST\_WAITS$$
$$= \frac{\sum_{all(a,b)_t} P(a,b)_t \cdot R_{req}(a,b)_t \cdot (P_{BLK}(a,b)_t + P_{DLK}(a,b)_t)}{\sum_{all(a,b)_t} P(a,b)_t \cdot R_{req}(a,b)_t}$$

요구연산 중에서 교착상태를 유발하는 확률 *PROBABILITY_A_REQUEST_DEADLOCKS*도 *DLK* 전이만 고려해서 같은 방법으로 계산한다.

마지막으로 교착상태로 형성되는 사이클의 길이가 2일 확률을 계산해 보자.

$$PROBABILITY\_OF\_LENGTH\_2\_CYCLE$$
$$= \frac{DL\_RATE\_2}{DL\_RATE}$$

*DL_RATE_2*는 <식9>에서 $P_{DLK}(a,b)_t$ 대신에 길이 2인 교착상태의 발생 확률 $P_{DLK2}(a,b)_t$를 계산하여 대입하면 된다. $P_{DLK2}(a,b)_t$는 <식5>에서 j 값을 1로 고정시켰을 때의 값으로부터 구할 수 있다.

V. 결 론

자원 할당 상태를 $(a,b)_t$로 표현하는 모델을 이용하여, 시스템이 안정 상태에 있을 때 교착상태의 평균 발생 주기와 요구연산이 보류될 확률 및 교착상태를 유발할 확률, 사이클의 길이가 2일 확률 등을 계산하였다.

교착상태의 특성을 해석적으로 분석하고자 했던 연구들이 대부분 프로세스나 자원의 수가 두 세 개 정도인 시스템에만 적용되었던 것과는 달리[1,3], 본 논문에서는 <식1>과 <식2>에서 보인 바와 같이 상태의 개수를 획기적으로 줄임으로써 보다 큰 규모의 범위까지 계산 가능한 식을 제시하였다. 또한 자원의 오류 및 복구 비율이 시스템에 미치는 영향도 아울러 분석 가능하도록 설계하였다.

앞으로 자원의 개수가 각 종류마다 여러 개인 시스템으로의 확장과, 요구연산과 반환연산이 다른 형태를 지닐 경우의 결과, 그리고 분산 처리 시스템에서의 적용 가능성들에 대해서 보다 연구되어져야 할 것이다.

참고문헌

[1] Ellis, C. A., "Probabilistic Models of Computer Deadlock," Report CU_CS_041_74, Dept. of Computer Science, Univ. of Colorado, Boulder, Colo., Apr. 1974

[2] Gray, J. N., et al., "A Straw Man Analysis of Probability of Waiting and Deadlocks," Research Report RJ3066, IBM Research Lab., 1981

[3] Koh, K. and Yoo, W., "A Probabilistic Model of Deadlock," 2nd Int'l Conference on Computer and Application, IEEE, Beijing, China, June 1987

[4] 고건, 성순용, "확률적 모델을 이용한 교착상태의 특성 분석," 정보과학회 논문지, 1991

[5] 성순용, 고건, "안정상태에서의 교착상태 확률모델과 Straw Man 분석과의 비교연구," 정보과학회 논문지, 1992

[6] Daniel I. A. Cohen, Basic Techniques of Combinatorial Theory, John Wiley & Sons, 1978

[7] L. Kleinrock, Queueing Systems, Vol. 1; Theory, Wiley-Interscience,1975

저자소개

성순용(Soonyong Seong)

1983 서울대학교 자연과학대학 계산통계학과 (이학사)

1985 서울대학교 대학원 계산통계학과(이학석사)

1992 서울대학교 대학원 계산통계학과(이학박사)

1989 - 현재 부산외국어대학교 컴퓨터공학과 교수

※관심분야 : 운영체제, 성능평가