

SLA 를 고려한 주-백업 웹 스위치의 가용도 분석

강창훈* 박기진** 김성수***
극동정보대학 방송영상미디어과* 아주대학교 공과대학 산업정보시스템공학부**
아주대학교 정보통신전문대학원
chkang@kdc.ac.kr* kiejin@ajou.ac.kr** sskim@ajou.ac.kr***

An Availability Analysis of Primary-Backup Web Switch Considering Service Level Agreement

Changhoon Kang* Kiejin Park** Sungsoo Kim***
Dept. of Visual Broadcasting Media, Keukdong College*
Division of Industrial & Information Systems Engineering, Ajou University**
Graduate School of Information and Communication, Ajou University***

요 약

클러스터 기반 웹 시스템에서 클라이언트의 계층별 요청 형태에 따라 차별화된 서비스를 제공하려는 시도가 활발하다. 서비스 제공자와 클라이언트간의 서비스 계약 조건(SLA: Service Level Agreement)을 만족시키기 위해, 웹 스위치 단계에서 클라이언트의 서비스 요청의 내용을 파악한 후 동적으로 서버에 요청처리 작업을 할당함으로써, 클라이언트 계층별 SLA 의 제공이 가능하다. 하지만, 모든 클라이언트의 요청은 웹 스위치를 통하여 각 서버에 분배되기 때문에, 웹 스위치가 내부적 결함이나 외부의 침입에 의해 정상적인 서비스를 제공하지 못하는 경우, 웹 시스템 서비스의 다운을 초래 할 가능성이 높다. 본 논문에서는, 웹 스위치의 내부적인 결함은 물론 외부의 악의적인 공격(예, 침입, 바이러스)으로 인하여 웹 스위치에 문제가 발생하는 경우 대기하고 있는 백업(backup) 웹 스위치로 대치하여 서비스를 계속 수행함으로써, SLA 가 고려된 웹 시스템의 가용도(Availability)를 높일 수 있는 구조를 제안하였고, 마코브(Marcov) 분석 및 실험을 통해 시스템 가용도와 SLA 척도를 정량적으로 평가하였다.

1. 서 론

최근 웹 서비스의 형태가 비즈니스 관련 미디어를 처리 하는 방향으로 발전되고 있어 클라이언트 계층별로 차별화된 서비스(Differentiated Service)를 제공할 필요성이 증대하고 있으며 차별화된 고품질의 서비스를 제공하기 위해서, 웹 서버에서 서비스 수준을 고려할 수 있어야 한다. SLA(Service Level Agreement)는 서비스 제공자와 대상 고객간 계약으로, 제공되어질 서비스와 그와 연관된 조건들을 사전에 약속하여 계약된 만큼의 서비스를 제공하는 것이며[1], 차별화된 서비스를 통해 고객과 서비스 제공자간의 SLA 의 보장이 가능하다.

한편 웹 상에서 차별화된 서비스를 제공하기 위하여 웹 서버 단계의 QoWS(Quality of Web Services) 개념이 제시 되었으며[2,3], 성능분리(Performance Isolation)방식을 통해 QoWS 를 보장하는 방식은, SLA 를 만족시키기 위해 클러스터를 구성하고 있는 서버마다 서로 다른 계층의 클라이언트 요청을 담당하게 하고 상위 계층의 클라이언트 요청일수록 우선적으로 서버를 할당하는 개념이다[4].

클라이언트의 요청을 처리할 서버를 결정하는 역할은 웹 스위치(Web Switch)가 담당하게 되는데, 웹 스위치가 하드웨어/소프트웨어적인 결함으로 인한 서비스 중단 또는 외부로부터의 악의적 공격 대상이 되어 원래 제공해야 하는 정상적인 서비스를 제공하지 못 할 경우, 결국에는

웹 시스템 전체의 다운을 초래할 가능성이 높다. 따라서 본 논문에서는 웹 시스템의 고장 가능성을 줄이기 위하여, 주 웹 스위치와 백업 웹 스위치를 두어, 주 웹 스위치에 문제가 발생하는 경우 대기하고 있는 백업(backup) 웹 스위치로 대치하여 서비스를 계속 수행함으로써, 웹 시스템의 가용도를 높일 수 있는 구조를 제안하였고, 마코브 분석을 통해 시스템 가용도를 정량적으로 평가 하였다. 본 논문의 2 장에서는 관련연구를 살펴보고, 3 장에서는 주-백업 웹 스위치 클러스터 시스템 모델을 제시하며, 4 장에서는 제시한 구조의 성능을 평가 하였고 5 장에서는 결론을 내렸다.

2. 관련 연구

클러스터 웹 시스템 환경에서 클라이언트 요청을 처리하기 위해, 웹 서버 단계에서 클라이언트 요청의 우선 순위에 따라 차별화된 서비스[5]를 제공할 수 있는 기법에 대한 연구가 이루어졌으며, 또한 클러스터를 구성하고 있는 서버들을 정적 혹은 동적으로 분할하는 기법이 연구되고 있다. 정적 분할 기법은 클라이언트의 계층별 요청 수준에 따라 고정된 수의 서버를 계층별로 할당하는 방법이다. 이는 특정 계층에 할당된 서버가 과부하 상태인 반면에 다른 계층의 서버는 유휴 상태에 있을 수 있는 단점이 있으며, 클라이언트의 계층별 요청 수가 자주 변하는 웹 환경에는 적당하지 않다[6].

클라이언트의 계층별 요청 수준에 따라 동적으로 서버를 할당하는 동적 분할 기법에는 DDS(Demand-driven Service

본 연구는 한국과학재단 목적기초연구(R05-2003-000-10345-1) 지원으로 수행되었음.

Differentiation)[7], Dynamic Partitioning[8] 기법 등이 있다. DDSO 기법은 클라이언트 요청량에 따라 CPU 와 디스크 I/O 용량을 할당함으로써 차별화된 서비스를 제공하고, e-비즈니스와 관련된 웹 사이트 콘텐츠처럼 CPU 와 디스크 I/O 의 처리를 많이 요구하는 동적 요구가 많은 상황에 적합하다. Dynamic Partitioning 기법은 SLA 를 만족하는 서비스를 상위 계층 사용자에게 제공하기 위해 서버의 부하량에 따라 서버를 동적으로 분할하는 기법이다. 그러나 이것은 정적 요구를 전혀 고려하지 않고 동적 요구만 고려했기 때문에 정적 요구가 많아질 경우 성능 저하가 발생하는 단점이 있다

LARD(Locality-Aware Request Distribution)[9] 기법은 캐쉬 개념을 적용한 방법으로 많이 사용되는 콘텐츠는 많은 요청이 이루어질 것으로 예상하여 서버를 할당하는 방식이다. 클라이언트의 요청을 두 개의 계층(High Set, Low Set)으로 분리하여 차별화된 서비스를 제공하려는 시도가 있었으며[8], [10]는 이 방법을 일반화시킨 개념으로, 클라이언트의 요청을 n 개의 계층으로 분할하여 각 요청들을 분배하였다.

기존의 연구들은 각 요청을 해당 서버로 분배해주는 웹 스위치가 안정적인 것이라는 가정을 적용하고 있다. 그러나 웹 스위치에 결함이 발생하게 되면 클러스터의 요청을 분배할 수 없게 되어, 해당 사이트는 서비스 불능 상태가 될 가능성이 높다. 따라서 본 논문에서는 클러스터 시스템의 웹 스위치가 내부적 결함이나 외부적인 침입(Intrusion)에 의해 기능이 정지되는 가능성을 줄이기 위하여, 주 웹 스위치에 이상이 발생하였을 때 백업 웹 스위치로 대체하여 정상적으로 스위치 동작이 이루어질 수 있는 SLA 가 고려된 주-백업 웹 스위치 부하 분배 클러스터 시스템의 구조를 제안하고, 성능을 분석하였다

3. 주-백업 웹 스위치 클러스터 시스템 모델(Primary-Backup Web Switch Cluster System Model)

3.1 주-백업 웹 스위치(Primary-Backup Web Switch)

그림 1 은 본 논문에서 고려한 웹 클러스터 시스템의 구조를 보여준다. 웹 스위치는 서비스 노드(Real Server)들 간에 작업들을 고루 분배하는 기능을 수행하며, 과중하게 부하를 받고 있는 서버들이 다른 서버들에게 작업을 전달하거나, 유휴 상태에 있는 서버의 작업 분배 요청을 처리한다[10].

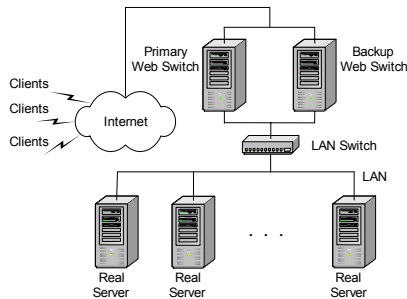


그림 1. 주-백업 웹 스위치 시스템

웹 스위치는 지속적으로 부하분배 서비스를 제공해야 하므로, 웹 스위치의 과도한 집중이나, 시스템 이상 혹은 외부의 침입으로 다운된 경우, 대기하고 있던 백업 웹 스위치가 주 웹 스위치의 역할을 대신 수행시킬 수 있어야 한

다. 백업 웹 스위치는 주 웹 스위치가 작동하지 않으면 대체되는 웹 스위치로, 주 웹 스위치와 연결되어 각각 Heartbeat 데몬을 가동시켜 주기적으로 서로 Heartbeat 을 주고 받으며 Heartbeat 을 일정 시간 내에 받지 못하면 상대방 웹 스위치가 다운된 것으로 판단한다.

그림 2 는 본 논문에서 고려한 주-백업 웹 스위치 시스템의 상태 전이도(State Transition Diagram)를 보여준다. 주-백업 웹 스위치 시스템의 주 웹 스위치와 백업 웹 스위치는 각각 S(Safe State)와 F(Failure State) 상태로 구성된다.

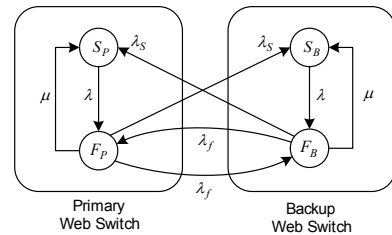


그림 2. 주-백업 웹 스위치 상태전이도

주 웹 스위치는 안정상태(S_P)에서 시스템 내부의 결함이나 외부의 공격 혹은 침입에 의해 λ 의 고장률로 시스템이 고장나게 된다. 또한 주 웹 스위치가 고장상태(F_P)에서 백업 웹 스위치가 안정상태(S_B)에 있다면 백업 웹 스위치로 대체되어 부하 분배의 기능을 수행하고, 자체 시스템이 수리되면 자체 시스템이 주 웹 스위치의 기능을 수행한다. 만약 백업 웹 스위치도 고장 상태에 있는 경우, 둘 중 하나 혹은 두 대 모두 수리될 때 까지 부하 분배 기능을 수행하지 못함으로 인해, 해당 클러스터 웹 서버가 고장 상태로 존재하게 된다. 본 논문에서 제안한 웹 스위치 시스템에 사용된 가정은 다음과 같다.

- 주-백업 웹 스위치는 Hot Standby 방식으로 대기 상태에 있다.
- 주 웹 스위치와 백업 웹 스위치의 평균 고장($1/\lambda$) 및 수리($1/\mu$) 시간 간격은 지수 분포를 따른다.
- 주 웹 스위치에서 백업 웹 스위치로 작업전이(Switchover) 시간은 평균값이 $1/\lambda_S$ 인 지수분포를 따른다.
- 임의의 웹 스위치가 고장인 상태(F_P 혹은 F_B)에서 또 다른 웹 스위치도 고장인 상태 즉, 동시에 고장나는 시간 간격은 평균값이 $1/\lambda_f$ 인 지수분포를 따른다.

평형상태(Steady State)에서의 균형식(Balance Equation)은 다음과 같다.

$$\lambda P_{S_P} = \mu P_{F_P} + \lambda_S P_{F_B} \tag{1}$$

$$\lambda P_{S_B} = \mu P_{F_B} + \lambda_S P_{F_P} \tag{2}$$

$$(\mu + \lambda_S + \lambda_f) P_{F_P} = \lambda P_{S_P} + \lambda_f P_{F_B} \tag{3}$$

$$(\mu + \lambda_S + \lambda_f) P_{F_B} = \lambda P_{S_B} + \lambda_f P_{F_P} \tag{4}$$

$$P_{S_P} + P_{F_P} + P_{S_B} + P_{F_B} = 1 \tag{5}$$

위의 균형식과 각 상태에서 머물 확률의 총합이 1 이 되는 식을 결합한 연립방정식을 풀면, 시스템이 평형일 때 각 상태에서 머물 확률을 얻을 수 있다.

$$P_{S_P} = \frac{\mu + \lambda_S}{2(\mu + \lambda_S) + \lambda} \tag{6}$$

$$P_{F_p} = \frac{\lambda}{2(\mu + \lambda_s) + \lambda} \quad (7)$$

$$P_{S_b} = P_{S_p} \quad (8)$$

$$P_{F_b} = P_{F_p} \quad (9)$$

3.2 서비스 분류(Service Classification)

서비스 분류는 로드발런서에 입력되는 클라이언트의 요청을 분류하여 계층별 큐(Queue)로 보내는 작업을 의미하며, 클라이언트 요청은 우선 순위에 따라 n 개의 클래스로 구분되며, 각각의 클래스를 담당할 서버에 할당된 클라이언트의 요청이 들어오면, 요청의 우선 순위에 따라 웹 스위치가 적합한 서버에 할당한다.

전체 서버 수가 N 대인, 서버 할당 방식은 식(10)과 같다. 여기서, $CS_i(t)$ 는 임의의 시간 t 에서 클래스 i 에 할당되는 서버의 수이며, $CS_{i-1}(t)$ 는 $CS_i(t)$ 보다 우선순위가 높다. 또한 $us_i(t)$ 는 임의의 시간 t 에서 클래스 i 에 속하는 클라이언트 요청을 처리하는 서버를 구분하는 경계값이다.

$$\begin{aligned} CS_1(t) &= \{1, \dots, us_1(t)\}, \\ CS_2(t) &= \{us_1(t)+1, \dots, us_2(t)\}, \\ &\vdots \\ CS_n(t) &= \{us_{n-1}(t)+1, \dots, N\}, \\ us_i(t) &\in \{1, \dots, N\}, \quad us_1(t) \geq 1, \quad us_{n-1}(t) \leq N-1, \quad us_i(t) \leq us_{i+1}(t) \end{aligned} \quad (10)$$

3.3 동적 분할(Dynamic Partition)

클러스터 시스템은 최초 가동시에 각각의 클라이언트 계층별로 고정적으로 서버를 할당하며, 한번 할당된 서버 노드는 클라이언트의 계층별 요청률이 변해도 수정되지 않는다. 식 (11)은 이러한 분할 방식을 나타내고 있으며, $us_i(0)$ 는 시스템 최초 가동시에 정해지는 값으로, 최초 웹 서버를 가동 시켰을 때 정해지는 i 번째 클래스를 서비스 하는 서버의 경계 값이다. 여기서 ρ_i 는 전체 요청 수와 i 클래스에 속하는 클라이언트의 요청 수의 비율이고, T_N 은 N 대의 서버 에서 허용 가능한 최대 지연 시간이다. 또한, $MaxConn(T_N)$ 은 지연 시간 T_N 을 만족시키는 최대 접속 수이며, T_i 는 i 번째 클래스의 SLA 를 고려한 임계 시간 (Threshold Time: $T_i < T_N$) 이다.

$$us_i(0) = \left\lceil \rho_i \cdot N \right\rceil \cdot \frac{MaxConn(T_N)}{MaxConn(T_i)}, \quad i=1, \dots, n-1 \quad (11)$$

동적 분할 기법은 상위 계층의 사용자의 SLA 를 만족하는 서비스를 제공하기 위해, 특정 클래스를 서비스하는 노드에 과부하가 걸렸을 경우, 낮은 클래스의 서버를 추가적으로 이용하고, 과부하 상황이 해소되면 다시 서버를 반납하는 개념으로, 부하 변화에 맞게 능동적으로 대처할 수 있다.

$$Diff = SumLoad_{CS_i}(t) - (us_i(t-1) - us_{i-1}(t-1)) \cdot MaxConn(T_i);$$

$$if(|Diff| > SumLoad_{CS_i}(t) \cdot \alpha)$$

if

$$(SumLoad_{CS_i}(t) > (us_i(t-1) - us_{i-1}(t-1)) \cdot MaxConn(T_i))$$

$$us_i(t) = us_i(t-1) + 1;$$

else

$$if(SumLoad_{CS_i}(t) < (us_i(t-1) - us_{i-1}(t-1)) \cdot MaxConn(T_i))$$

$$us_i(t) = us_i(t-1) - 1;$$

$$\} \quad (12)$$

식 (12)는 계층 i 를 서비스 하는 서버의 동적 분할 방식을 표시하고 있으며, 부하의 변동 비율(α)에 따라 클라이언트 계층별 서버 노드의 경계값이 계산된다. $SumLoad_{CS_i}(t)$ 는 $CS_i(t-1)$ 에서의 서버 부하(Load)의 합이며, 부하는 클라이언트의 최대 접속 수($MaxConn$)를 의미한다.

4. 실험 및 분석

두 개의 웹 스위치가 모두 작동하거나 한 대의 웹 스위치가 작동하는 경우 웹 스위치 시스템은 가용하다고 볼 수 있다. 즉, 두개의 웹 스위치가 모두 고장난 경우를 제외하면 가용한 상태라고 볼 수 있다. 따라서 본 논문에서 제시한 시스템의 가용도 정의는 다음과 같다.

$$Availability = 1 - (P_{F_p} + P_{F_b}) \quad (13)$$

또한 본 논문에서 사용한 시스템 파라미터[11]는 표 1 과 같다. 웹 스위치의 고장(λ) 은 6 개월에 1 회, 고장 수리에 2 시간이 소요되며, 웹 스위치를 대체하는 작업전 시간은 10 초가 소요된다. 또한 동시에 두 서버가 고장나는 경우는 2년에 1회이다.

표 1 시스템 운영 파라미터

고장률 (λ)	2 time / year	1/(6*30*24*3600)
수리율 (μ)	1 time / 2 hours	1/(2*3600)
작업전이율 (λ_s)	1 time / 10 sec.	1/(10)
동시고장률 (λ_f)	1 time / 2 years	1/(24*30*24*3600)

그림 3 은 한 개의 웹 스위치를 사용한 구조와 주-백업 웹 스위치 구조간 고장률 변화에 따라 가용도를 측정한 결과이다. 한 개의 웹 스위치 사용시 고장이 자주 일어나면 (즉, 고장 시간 간격이 작을 때) 가용도가 현저히 떨어지고, 고장률이 낮아질수록 가용도가 증가하는 반면, 주-백업 웹 스위치 모델은 고장률 변화에 거의 무관하게 가용도가 1 에 가까워(0.999998) 항상 가용한 상태라고 볼 수 있다. 이런 결과는 주-백업 웹 스위치 구조는 현대의 웹 스위치 고장시 다른 여분의 웹 스위치로 대체되기 때문이다.

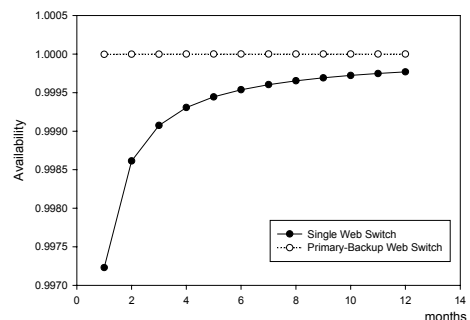


그림 3 고장률 변화에 따른 가용도

그림 4 는 수리율에 따른 가용도 변화로써 한 개의 웹 스위치인 경우는 수리 시간이 많이 걸릴수록 가용도는 떨어지는 반면, 주-백업 웹 스위치 구조는 수리 시간에 거의 무관하게 가용도를 유지하는데(0.9999994), 이는 현대의 웹 스위치가 수리되는 동안에 여분의 백업 스위치가 부하 분배 기능을 계속적으로 수행하기 때문이다.

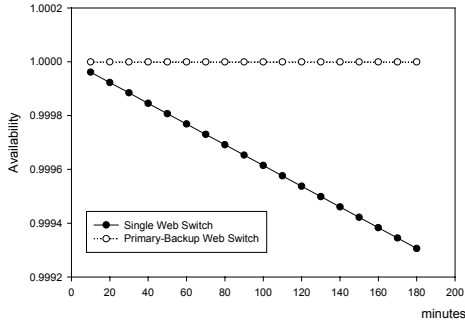


그림 4 수리율 변화에 따른 가용도

실제적인 웹 서버 클러스터 시스템의 성능 분할을 원활히 시뮬레이션 하기 위해, 클라이언트의 도착률에 따른 응답시간 및 승인 거절된 요청 수에 관하여 분석하였으며, 95-percentile은 SLA를 고려하기 위해 사용된 척도로써, 요청중 95% 이상은 서비스 제공자와 고객간에 계약된 시간 안에 응답시간을 만족시킴을 의미한다. 시뮬레이션에 사용된 웹 서버 클러스터 시스템의 운영 파라미터는 표 2, 3과 같다[7].

표 2. 정적·동적 요청 부하 모델

Type	Mean service time	Frequency
Dynamic requests	700 msec.	0.20
Static requests	100 msec.	0.80

표 3. 시스템 파라미터 (단위:second)

서비스 요청 도착 시간 간격	Exp(평균 서비스 시간)
서버 대수	10
클래스별 요청 도착 비율 $class_H, class_M, class_L$	[0.2, 0.3, 0.5]

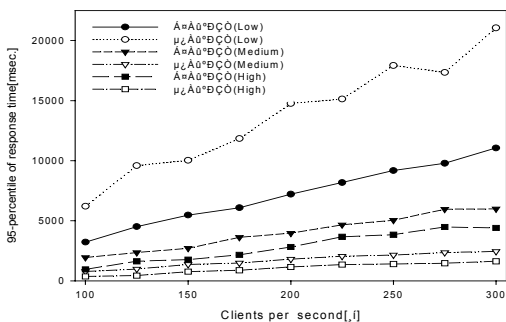


그림 5. 도착률에 따른 분할기법 간의 응답 시간 비교

그림 5는 3개의 클래스(High, Medium, Low)를 대상으로, High 클래스와 Medium 클래스, Low 클래스를 정적·동적

분할에 따른 응답 시간을 클라이언트의 도착률의 변동에 따라 표시하였다. 정적 분할보다는 동적 분할 기법의 응답 시간이 모든 클래스에서 작게 나왔으며, 이는 동적 분할 기법이 사용자 요청을 부하 변화에 맞게 대처하였기 때문이다.

5. 결론

SLA 를 지원하기 위한 클러스터 웹 시스템에서 웹 스위치는 클라이언트로부터 모든 요청을 받아들여 실제 서버들에게 부하를 균형있게 분배하는 역할을 담당한다. 따라서 웹 스위치의 역할이 매우 중요하고, 웹 스위치가 고장이 발생하면 해당 클러스터 시스템 전체가 고장으로 인식된다. 본 논문에서는 웹 스위치를 두 대로 운영하는 주-백업 웹 스위치 시스템 구조를 제시하였고, 가용도를 분석하였다. 그 결과 한 개의 웹 스위치를 운영하는 것보다 주-백업 웹 스위치 구조를 채택할 경우 가용도를 매우 높이고 일정하게 유지할 수 있음을 확인하였다. 추후에는 SLA 를 보장하기 위하여 사용자 계층을 다중계층(Multiclass)로 확장한 서버 분할 기법 및 웹 클러스터의 로드밸런싱을 위한 고가용성, 신뢰성에 대한 연구를 수행할 계획이다.

참고문헌

- [1] J. Pu, et al., "SLA Admission Controller for Reliable MPLS Networks," Applied Informatics, pp. 630-635, 2003.
- [2] V. Cardellini, et al., "Enhancing a Web-server Cluster with Quality of Service Mechanisms," Proceedings of 21st IEEE International Performance, Computing, and Communications Conference, pp. 205-212, 2002.
- [3] V. Cardellini, et al., "A performance study of distributed architectures for the quality of Web services," Proc. of Hawaii Int'l Conf. on System Sciences (HICSS-34), Software Technology Track, Maui, Hawaii, IEEE Computer Society, pp. 3551-3560, 2001.
- [4] M. Aron, et al., "Cluster reserves resource management in cluster-based network servers," In Proc. of ACM Sigmetrics 2000, pp. 90-101, 2000.
- [5] N. Vasiliou, et al., "Providing a differentiated quality of service in a World Wide Web server." SIGMETRICS Performance Evaluation Review, Vol. 28, No. 2, pp. 22-28, 2000.
- [6] V. Cardellini, et al., "Web switch support for differentiated services." SIGMETRICS Performance Evaluation Review, Vol. 29, No. 2, pp. 14-19, 2001.
- [7] H. Zhu, et al., "Demand-driven service differentiation in cluster-based network servers," In Proc. of IEEE Infocom 2001, pp. 679-688, 2001.
- [8] M. Andreolini, et al., "A Cluster-Based Web System Providing Differentiated and Guaranteed Services," Cluster Computing, Vol. 7, No. 1, pp. 7-19, 2004.
- [9] V. Pai, et al., "Locality-aware Request Distribution in Cluster-based Network Servers," In Proceedings of 8th ACM Conference on Architecture Support for Programming Languages, pp. 205-216, 1998.
- [10] 고현주 외 2 인, "SLA 를 고려한 웹 서버 부하 분산 기법." 한국정보과학회 가을학술발표논문집, Vol. 31, No. 2, pp. 652-654, 2004.
- [11] Y. Huang, et al., "Software Rejuvenation: Analysis, Module and Applications," IEEE Intl. Symposium on Fault Tolerant Computing, FTCS 25, pp. 381-390, June 1995.