대기시간 데드라인 조건을 고려한 (n,k)-way 시스템의 가용 성능 분석

(Analysis of Available Performance Satisfying Waiting Time Deadline for (n,k)-way Systems)

박 기 진 † 김 성 수 ††
(Kiejin Park) (Sungsoo Kim)

요 약 일정 수준의 시스템 성능을 제공하기 위해 다수의 서버를 클러스터로 연결하여 동시에 가동할 경우, 서버의 가동 대수가 증가함으로 인해 발생하는 가용도 저하 문제를 해결해야 하며, 이를 위해서는 시스템의 성능 변화를 반영할 수 있는 가용 성능에 대한 명확한 정의가 요구된다. 기존의 가용도에 관한 연구들은 시스템의 성능(대기시간, 응답시간 등)을 고려하지 않는 가용도 계산 모델이 주를 이루고 있으며, 또한 클러스터 시스템을 구성하는 주서버의 결함 발생을 허용하기 위한 방안에 대한 연구가 부족하였다. 본 논문에서는 위와 같은 문제점들에 대한 해결 방안의 일환으로, 비용 효율적인 결함허용을 위해 n 대의 주서버와 k 대의 여분서버로 구성된 클러스터 시스템에서 대기시간을 일정 수준이하로 만족시키는지에 근거한 새로운 가용도 척도를 제시하였으며, 다양한 시스템 운영 상태에 대한 실험을 통하여, 시스템의 대기시간 데드라인 성능이 포함된 보다 현실적인 가용도 및 downtime을 계산하였다.

키워드 : (n,k)-way 시스템, 가용도, 클러스터, 그리드, 결함 허용

Abstract As cluster systems used for high performance computing consist of large number of running servers, one has to solve the low availability problems occurred by the high chance of the server failures. To handle the problems, it is necessary to have the precise definition of available performance of cluster systems that represents availability and performability of the systems simultaneously. Previous research results that mention availability issues lack for concerning system performance such as waiting time and response time in their availability definition. In this paper, we propose a new availability metric for (n,k)-way cluster systems which compose of n primary servers and k backup servers. With the metric, the change of system performance according to arrival rates is captured and the waiting time of a request can be kept below to a certain level. Using various system operating parameters, we calculate availability and downtime of cluster systems along with waiting time deadline.

Key words : (n,k)-way System, Availability, Cluster, Grid, Fault Tolerance

1. 서 론

일반적으로 컴퓨터 시스템의 가용도와 성능을 동시에 분석하기는 어렵다[1,2]. 이유는 가용도 모델 및 계산에서 사용되는 평가척도의 시간 단위 및 평가 대상이 되는 개체의 수는, 성능 평가에 사용되는 시간 단위 및 개체의 수에 비해 비율적으로 상당히 적기 때문이다. 예를 들면, 가용도 평가를 위해, 대상 시스템의 고장은 1년에 한번씩 발생한다고 가정하고, 단위 시간(초) 간격 당 도착하는 서비스 요청회수는 1회라고 했을 때, 두 척도의 수치상 비율은 1 : 31,536,000이다(3600 *sec* * 24 *hour* * 365 *day*). 컴퓨터 시스템의 수행 성능(performability)을 고려한 가용도 분석 문제를 해결할 경우, 다양한 수준의 가용 능력 및 성능을 달성하기 위한 컴퓨터 시스템 구조 및 상위 수준의 디자인 단계에서 사용될 수 있는 정보를 얻을 수 있는 장점이 있다.

컴퓨터 시스템의 성능 변화를 반영할 수 있는 가용도

· 본 연구는 한국과학재단 목적기초연구(R05-2003-000-10345-0) 지원으로 수행되었음.
· 이 논문은 2003년도 두뇌한국21사업에 의하여 지원되었음
† 종신회원 : 안양대학교 컴퓨터학과 교수
kiejin@aycc.anyang.ac.kr
†† 종신회원 : 아주대학교 정보통신전문대학원 교수
sskim@ajou.ac.kr
논문접수 : 2001년 4월 25일
심사완료 : 2003년 6월 19일

분석 문제를 해결하기 위해서는 시스템 가용도에 대한 명확한 정의가 요구된다. N대의 서버로 구성된 N-way 클러스터 시스템에서[3-6] "N 대의 서버 중에 1대의 서버만 동작하고 있더라도 시스템이 가용하다"라고 말할 수 있는지, 혹은 "클러스터 시스템내의 서버가 몇 대가 살아 있던 간에 어느 순간에 접속하고 있는 사용자가 일정시간 이내에 답변을 얻으면 가용하다"라고 이야기 할 수 있는 것인지, 또는 "클러스터 시스템의 이용률이 일정 수준 이하이면 이를 가용하다"고 할 수 있는지를 고려해 볼 필요가 있다.

- 클러스터 시스템을 구성하고 있는 서버 중 최소한 한 대 이상의 서버가 작동할 경우, 이를 가용하다고 정의한 경우는 이미 저자들의 논문[2]에서 다루었으며, 대기시간(waiting time), 응답시간(response time) 등의 성능 척도를 고려해야 사용자가 느끼는 정확한 가용도를 계산할 수 있다고 지적한 바 있다.
- 응답 시간을 가용도 평가 척도에 사용한 연구 결과[1]가 있으나, 가용도 모델링에서 가정한 시스템 상태 변수를 서비스 요청 트랜잭션의 수로 했기 때문에 현실적으로 대용량 트랜잭션을 다루는 클러스터 시스템의 정확한 성능을 평가하기는 곤란하다. 하지만 기존의 가용도 정의의 문제점을 극복하기 위해 응답시간 개념을 적용한 것은 바람직한 시도라고 볼 수 있다.
- 이용률을 시스템 성능 변수로 고려할 경우, 시스템 이용률의 계산 특성상 일정 수준이상이 되면 급속히 증가하는 속성상 사용하기 힘들 것으로 판단된다.

대량 트랜잭션을 처리해야 하는 클러스터 시스템은 빈번한 통신 두절과 데이타의 복잡성으로 인하여, 시스템의 결함 발생 가능성이 더욱 높으며, 이로 인해 시스템 서비스의 오동작 또는 수행 중단으로 이어지는 사례가 늘어나고 있다[7-9]. 한편 요구되는 시스템 성능을 확보하기 위해 다수의 서버를 클러스터로 연결하여 동시에 가동할 경우, 서버의 가동 대수가 증가함으로 인해 발생하는 가용도 저하 문제가 파생된다[10,11]. 그림 1은 본 논문에서 분석하고자 하는 (n,k)-way 클러스터 시스템의 연결구조 모습이다. (n,k)-way 시스템은 n 대의 주서버(primary server: P_i, i=1,2,...,n)와 k 대의 여분서버(backup server: Bi, i=1,2,...,k)로 구성되어 있으며, 임의의 주서버에 고장이 발생할 경우, k 대의 여분서버 중의 한 대가 주서버의 역할을 대신하는 시스템을 말한다[12-14]. 시스템 서비스에 필요한 실제 데이타는 storage interconnect(fiber channel, TCP/IP, SCSI 등)에 연결된 모든 저장 장치로부터 서버와 공유되며, 서비스를 요청하는 클라이언트는 초고속 통신망을 통해 클러스터 서버와 연결되는 구조를 가지고 있다.

지금까지의 컴퓨터 시스템 가용도에 관한 연구들은

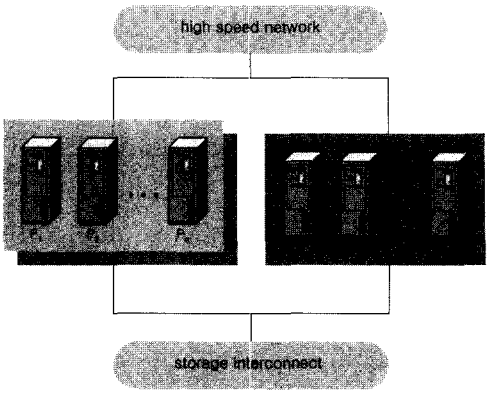

그림 1 (n,k)-way 클러스터 시스템의 연결 구조

시스템의 성능(대기시간, 응답시간 등)을 고려하지 않는 가용도 계산 모델이 주를 이루고 있으며[15-17], 또한 클러스터 시스템 주서버의 결함을 허용하는 일정 수준의 가용도를 확보하기 위한 비용 효율적인 방안에 대한 연구가 부족하였다[18,19]. 본 논문에서는 위와 같은 문제점들에 대한 해결 방안의 일환으로, 비용 효율적인 결함허용을 위해 n 대의 주서버와 k 대의 여분서버로 구성된 클러스터 시스템에서 대기시간을 일정 수준이하로 만족시키는 즉 대기시간 데드라인 개념(waiting time deadline)이 포함된 새로운 가용도 척도를 제시하였다. 또한 주서버의 고장 발생시 여분서버로의 작업전이 상태를 클러스터 시스템 모델링에 포함시켰으며, 작업전이 상태에서 머무는 시간을 지수분포 대신에 r-stage Erlangian 분포를 사용하여 확정시간(deterministic time)을 표현할 수 있도록 하였다. 본 논문의 서론 부분에서는 관련 연구 결과와 문제를 정의하였으며, 2장에서는 고객 도착률에 따른 대기시간 데드라인을 기준으로 한 (n,k)-way 클러스터 시스템의 가용도 모델 및 새로운 가용도 평가 척도를 정의하고, 3장에서는 제안된 모델의 수학적 해석 방법의 정확성을 검증하기 위한 다양한 시스템 운영 조건에 대한 실험을 수행하였다. 마지막으로 결론 부분에서는 제안된 분석 기법의 활용 방안 및 향후 연구에 관하여 논하였다.

2. (n,k)-way 클러스터 시스템 모델

다수의 서버 가동으로 인한 가용도 저하 문제를 해결하기 위해 (n,k)-way 클러스터 시스템을 분석 대상으로 하였으며, 일반적으로 (n,k)-way 시스템 분석에서는 여분 하드웨어의 수를 최소화하면서 동시에 원하는 시스템의 신뢰 척도를 확보하는 것을 목표로 삼는다. 작업전이 시간의 길이에 따라 구분되고 있는 여분서버로의 작업전이 방식 중, Active/Standby 방식은 Active/Active

방식에 비해 비용이 저렴하며, 경성 실시간 제약 및 mission-critical 응용 분야가 아닌 웹 서비스 분야에 적합하다고 볼 수 있다[3]. 한편 대부분의 클러스터 시스템은 어느 한 서버가 동작을 멈추더라도 동일 클러스터내의 다른 서버에는 영향을 미치지 않고, 주서버의 고장 시에 여분서버로의 작업전이가 일정 시간 내에 이루어져야 하는 시스템 요구 사항을 포함하고 있으며, 클러스터 서버간의 작업 부하는 부하 균형기(load balancer)에 의해 제어되고, 요청된 작업은 도착 순서대로 각 서버에게 분배된다. 또한 대부분의 기존 연구 논문에서는 클러스터 서버의 고장 및 수리 시간 간격과 서비스 요청의 도착 시간 및 서비스 시간 간격은 지수 분포를 가정하고 있다. 이상에서 언급한 내용으로부터 (n,k)-way 클러스터 시스템 가용도 모델링에 사용될 기본적인 가정들을 정리하면 다음과 같다[1-3].

- n 대의 주서버가 클러스터로 연결되어 있으며, 각 서버는 서로 독립적으로 작업을 수행한다.
- k 대의 여분서버가 존재하며($n \geq k$), 임의의 주서버에 고장이 날 경우 이의 역할을 Active/Standby 방식으로 대체한다.
- 주서버와 여분서버의 고장률(λ), 수리율(μ)은 상수이다. 즉 고장 및 수리 시간 간격은 지수분포를 따른다.
- 여분서버는 주서버 고장 시에만 사용되며, 고장난 서버는 수리된다.
- 주서버에서 여분서버로 작업전이 시에 일정 시간이 소요되며, 작업전이 시간은 평균값이 $1/\lambda_s$인 r-stage Erlangian 분포를 따른다.
- 클러스터내의 주서버간의 작업 부하는 균등하며, 작업의 도착률(α)과 서비스률(β)은 상수이다. 즉 도착 및 서비스 시간 간격은 지수분포를 따른다. 또한 요청된 작업은 선입선출(FIFO)방식에 따라 서비스된다.

도착률의 증가(서비스 요청의 증가)로 인하여 시스템에서의 대기시간이 길어지게 되면, 클러스터 시스템을 구성하는 서버의 수학적 계산에 의한 가용도와 사용자들이 실제로 느끼는 가용도와는 많은 차이가 발생하게 된다. 서론에서 언급한 바와 같이 시스템 부하를 고려한 고객 도착률에 따른 대기시간이 포함된 가용도 평가 척도를 계산하기 위해서, 분할(decomposition) 기법을 적용하였다.

- 상위 단계에서는 클러스터 시스템의 고장-수리 상태 전이 모델링으로부터 성능이 포함되지 않는 기초적인 가용도 정보를 계산하였으며, 이 때 주서버의 고장으로 인한 여분서버로의 작업전이(switchover) 시간을 memoryless 성질의 지수 분포 대신에 r-stage Erlangian 분포로 가정하여, semi-Markov 과정으로 모델링함으로써, 현실에 보다 근접한 가용도 계산이 가능하도록 하였다.
- 하위 단계에서는 상위 단계에서 구해진 가용도 정보를 이용하여, 도착률에 따른 시스템의 대기시간 데드라인 성능을 고려한 새로운 가용도 평가 척도를 적용하여 시스템의 가용도를 계산하였으며, 이를 위해 M/M/c 큐잉 기법[20,21]을 적용하여 시스템에서의 대기시간이 일정 수준 이하인 조건을 만족하도록 하였다.

2.1 시스템 부하를 고려하지 않은 가용도 분석(상위 단계)

Active/Standby 작업전이 가동 방식에 따른 (n,k)-way 클러스터 시스템의 상태 모델을 그림 2에 나타내었다. 정상 상태(normal state)에서 가동되고 있는 서버는 (n,k), $(n,k-1)$, ..., $(n,0)$, $(n-1,0)$, ..., (1,0) 등의 가동중인 (주서버, 여분서버)의 수를 상태 변수로 가지고 있으며, k 개의 여분서버가 모두 사용될 때까지는 시스템의 가동 서버는 n 개로 유지된다. (0,0) 상태는 주서버와 여분서버의 가동이 모두 중지된 경우로 고장상태를 표시한다. 주서버에 고장이 발생할 경우 여분서버에서 작업을 대체 수행하기 위한 작업전이 상태를 회색의 겹쳐진 원으로 표시된 S_k, S_{k-1}, ..., S_1로 나타냈으며, 한편 여분서버에 고장이 발생할 경우 작업전이 시간이 불필요하므로 지연시간 없이 상태가 변한다. 그림 2의 모델링은 작업전이 상태에서 머무는 시간(sojourn time)의 분포가 memoryless 성질을 만족하지 않기 때문에, semi-Markov 프로세스 문제로 분류되며, 정확한 해를 구하는 간단한 방법이 존재하지 않는다. 고가용성 클러스터 시스템은 주서버에서 여분서버로의 작업전이가 일정 시간 이내에 이루어져야 하는 조건을 만족시켜야 하며, 이와 같이 현실적인 고가용성 클러스터 시스템의 운영 조건을 반영하는 모델의 수학적 해를 구하기 위해서, 그림 2에서 회색으로 칠해진 임의의 작업전이 상태 S_i

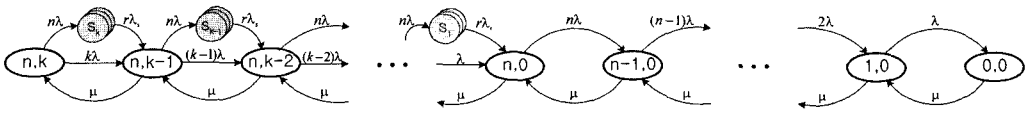

그림 2 (n,k)-way 클러스터 시스템의 고장-수리 상태 전이도

그림 3 작업전이 상태 S_i의 분할 상태 전이도

를 r개의 부분작업전이 상태(sub-switchover state: $S_{i,j}$)로 나눈 분할 상태 전이도로 모델링 하였다(그림 3 참조).

각각의 부분작업전이 상태($S_{i,j}$)에서의 작업전이 소요시간은 평균 $1/r\lambda_s$인 지수분포를 따른다. 지수분포의 합으로 이루어진 r-stage Erlangian 분포의 성질 중 상태 분할 개수(r)가 증가함에 따라 확률 분포의 분산 값이 점점 작아져서 결국에는 unit impulse 값을 가지는 사실을 적용하여[20], 본 모델링에서 요구되는 작업전이 상태에서의 확정 값을 가지는 sojourn time 문제를 해결하였다. 그림 2의 모든 상태에서 입력과 출력 비율이 일정해지는 평형(steady-state)일 때의 균형 방정식(balance equation)은 다음과 같다.

$$(n+k)\lambda P_{n,k} = \mu P_{n,k-1}$$

$$((n+j)\lambda+u)P_{n,j} = \mu P_{n,j-1} + (j+1)\lambda P_{n,j+1} + r\lambda_s PS_{j+1,r},$$

$$j = 1,2,\ldots,k-1$$

$$(n\lambda+u)P_{n,0} = \mu P_{n-1,0} + \lambda P_{n,1} + r\lambda_s PS_{1,r}$$

$$(i\lambda+u)P_{i,0} = \mu P_{i-1,0} + (i+1)\lambda P_{i+1,0} \quad ,i = 1,2,\ldots,n-1$$

$$\mu P_{0,0} = \lambda P_{1,0}$$

$$r\lambda_s PS_{i,1} = n\lambda P_{n,i} \quad ,i = 1,2,\ldots,k$$

$$r\lambda_s PS_{i,j} = r\lambda_s PS_{i,j-1} \quad ,i = 1,2,\ldots,k \quad ,j = 2,3,\ldots,r$$

$$\sum_{i=0}^{n-1} P_{i,0} + \sum_{j=0}^{k} P_{n,j} + \sum_{i=1}^{k}\sum_{j=1}^{r} PS_{i,j} = 1 \quad ; \text{conservation 방정식}$$

각 상태 확률의 의미는 아래와 같다.

$P_{n,j}$: 시스템이 평형일 때, 주서버가 n 대이고 여분서버의 수가 j 대일 상태 확률

$P_{i,0}$: 시스템이 평형일 때, 주서버가 i 대이고, 여분서버가 0 대일 상태 확률

$PS_{i,j}$: 평형 시스템의 여분서버가 i 대이고, 주서버의 고장으로 인한 작업전이 시에 추가로 거쳐야 할 부분작업전이 상태의 개수가 $r-j$ 개 남아 있을 확률

위의 균형 방정식과 각 상태에서 머물 확률의 총합이 1이 되는 conservation 방정식을 결합한 연립 방정식을 풀면, 시스템이 평형일 때, 각 상태에 머물 확률을 얻을 수 있다.

$$P_{n,k} = \left[1+(n+k)!\left(\sum_{i=0}^{k-1}\left(\frac{\lambda}{\mu}\right)^{k-i}\frac{1}{(n+i)!} + \sum_{i=0}^{n-1}\left(\frac{\lambda}{\mu}\right)^{n+k-i}\frac{1}{(n-i)!} + \frac{n\lambda}{\lambda_s}\sum_{i=1}^{k}\left(\frac{\lambda}{\mu}\right)^{k-i}\frac{1}{(n+i)!}\right)\right]^{-1},$$

$$n \geq k \quad ,k = 1,2,\ldots \tag{1}$$

$$P_{n,0} = \left[1+n!\sum_{i=0}^{n-1}\left(\frac{\lambda}{\mu}\right)^{n-i}\frac{1}{(n-i)!}\right]^{-1} \quad ,n = 1,2,\ldots \tag{2}$$

$$P_{n,j} = \left(\frac{\lambda}{\mu}\right)^{k-j}\frac{(n+k)!}{(n+j)!}P_{n,k} \quad ,j = 0,1,\ldots,k-1 \tag{3}$$

$$P_{i,0} = \left(\frac{\lambda}{\mu}\right)^{n-i}\frac{n!}{(n-i)!}P_{n,0} \quad ,i = 0,1,\ldots,n-1 \tag{4}$$

$$PS_{i,j} = \frac{n\lambda}{r\lambda_s}P_{n,i} \quad ,i = 1,2,\ldots,k \quad ,j = 1,2,\ldots,r \tag{5}$$

즉 시스템 운영 파라미터를 이용하여 $P_{n,k}$와 $P_{n,0}$을 계산하면, 그 외의 모든 정상 상태($P_{n,j}$, $P_{i,0}$), 및 작업전이 상태($PS_{i,j}$)에서의 확률을 차례로 계산할 수 있다. 클러스터 시스템을 구성하는 서버의 가동 여부만을 고려하여 가용도를 정의하면 아래와 같다.

기존 가용도:

$$A = \sum_{i=1}^{n-1} P_{i,0} + \sum_{j=0}^{k} P_{n,j} + \sum_{i=1}^{k}\sum_{j=1}^{r} PS_{i,j} = 1 - P_{0,0} \tag{6}$$

위 식은 모든 서버가 가동을 중지한 경우를 제외한 모든 상태에서 시스템의 서비스가 가용하다는 의미이다. 하지만 서버가 n 대 가동할 경우와 서버가 한 대 가동할 경우에, 시스템이 동일하게 가용하다고 하는 해석에는 문제가 있으며, 보다 현실적인 가용도 척도의 정의가 요구된다.

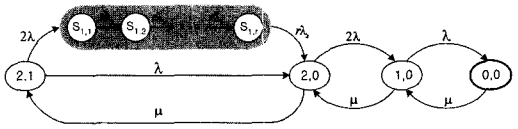

그림 4 (2,1) 클러스터 시스템의 고장-수리 상태 전이도

그림 4는 주서버가 두 대이고, 여분서버가 한 대인 (2,1) 클러스터 시스템의 상태 모델의 예이며 시스템이 평형일 때의 각 상태 확률은 다음과 같다.

1) 정상 상태 : $P_{2,1} + P_{2,0} + P_{1,0}$

2) 작업전이 상태 : $\sum_{j=1}^{r} PS_{1,j}$

3) 고장 상태 : $P_{0,0}$

가동중인 서버는 $i*\lambda$ (i: 서버 수)의 변화율로 고장이 발생하며, 고장난 서버는 μ의 수리률로 복구된다. 주서버 두 대와 여분서버 한 대의 가동이 모두 중지된 고장

상태(0,0)에서는 시스템 이용이 불가능하게 되며, 주서버의 고장으로 인해 발생한 여분서버로의 작업전이는 작업전이 상태($S_{1,j}$)를 통해 이루어진다. 한편 여분서버가 모두 사용되었을 경우((2,0),(1,0))에는 작업전이 상태가 모델링에 포함되지 않는다.

2.2 대기시간 데드라인을 고려한 가용 성능 평가(하위 단계)

서비스를 요청한 후, 서비스를 받기 이전까지 작업 큐에서 대기하고 있는 총 시간에 대한 평가가 사용자에게 더욱 현실적인 가용도 판단 기준이 될 수 있다. 시스템의 대기시간 데드라인 성능을 포함한 가용도를 정의하기 위하여, 고객의 서비스 요청과 서버의 가동 대수의 변동에 따른 서비스 대기시간 분포 함수를 가용도 정의에 반영하고자 한다. 즉 W를 큐에서 보낸 대기시간을 나타내는 랜덤 변수라고 할 때에, 서비스 요청에 대한 대기시간 함수($W_q(t)$)로부터, 큐에서 대기시간이 일정 시간(d)을 초과하지 않을 확률(acceptance ratio)이 Φ 이상인 조건을 표시하면 다음과 같다.

$$W_q(d) = pr\{W \le d\} \ge \Phi$$

또한 사용자가 시스템이 가용하다고 판단할 수 있는 대기시간의 최대값 즉 대기시간 데드라인을 D^*라고 했을 때, i 대의 주서버가 가동중인 클러스터 시스템의 가용 유무를 판정하기 위한 indicator 변수 X_i (i = 1,...,n, max(n,1), max(n,2), ..., max(n,k))의 값을 다음의 기준으로 결정한다.

$$d \le D^* : X_i = 1, \quad d > D^* : X_i = 0$$

(n,k)-way 클러스터 시스템의 주서버간에 작업부하가 균등하고, 트랜잭션의 도착 시간 간격과 시스템의 서비스 시간 간격이 지수 분포를 따르며, 요청된 작업은 선입선출 방식으로 처리된다는 가정에 따라서, 본 연구에서 요청되는 대기시간 분포는 M/M/c 큐잉 기법을 적용하여 구할 수 있다. M/M/c 시스템 큐에서의 대기시간 분포는 다음과 같다[21].

$$W_q(t) = \begin{cases} 1 - \dfrac{c\rho^c}{c!(c-\rho)} p^* & ,t = 0 \\ \dfrac{\rho^c (1 - e^{-(\beta c - \alpha)t})}{(c-1)!(c-\rho)} p^* + W_q(0) & ,t > 0 \end{cases} \quad (7)$$

단, $\rho = \alpha/\beta$: 이용률, $p^* = \left[\sum_{i=0}^{c-1} \frac{1}{i!} \rho^i + \frac{1}{c!} \rho^c \left(\frac{c\beta}{c\beta - \alpha} \right) \right]^{-1}$: 시스템에 작업이 한 개도 없을 확률

(n,k)-way 클러스터 시스템에서 여분서버가 없을 상태 확률($P_{i,0}$), 여분 서버가 있을 상태 확률($P_{n,j}$) 및 여분서버에서 주서버로의 작업전이 상태 확률($PS_{i,j}$)과 가용성을 판단하기 위한 변수 X_i와의 곱에 의해 결정되는 새로운 가용도 평가 척도, 즉 고객의 대기시간 데드라인 성능이 고려된 고객 도착률(α)에 따른 가용도는 아래와 같다(식 (8) 참조). 또한 주서버의 수(n)가 최소 필요 대수(m_0, 단 $m_0 \ge \lceil \rho \rceil$, ρ는 도착률과 서비스률의 비율을 나타내는 이용률) 이상이며, 여분서버가 없을 경우(k=0)에는 작업전이 상태를 가용도 계산에서 고려할 필요 없으므로, 작업전이 상태에 있을 확률은 가용도 계산에서 제외된다. 대기시간이 무한대로 길어지는 시스템 발산시, 즉 주서버의 가동 대수 $i = 0, 1, \ldots, m_0 - 1$일 때 $X_i = 0$이 된다.

새로운 가용도:

$$A_\alpha = \begin{cases} \sum_{i=m_0}^{n-1} X_i P_{i,0} + X_n \sum_{j=0}^{k} P_{n,j} + X_{n-1} \sum_{i=1}^{k} \sum_{j=1}^{r} PS_{i,j} & ,n > \mathrm{m}_0 \\ X_n P_{n,0} & ,\mathrm{n} = \mathrm{m}_0 \text{ and } \mathrm{k} = 0 \\ 0 & ,\mathrm{n} < \mathrm{m}_0 \end{cases} \quad (8)$$

서버의 고장, 수리 및 작업전이 항목만을 고려한 상위 단계의 분석을 통해 얻은 평형 상태 확률과 고객 도착률과 시스템 서비스율에 따른 대기시간 분포함수로부터 마감 시간에 근거한 새로운 가용도(즉 마감 시간내에 서비스가 가능한 경우)를 계산하는 과정은 다음과 같다.

• 대기시간 데드라인에 근거한 가용도 계산 알고리즘

Step 1: 서버의 고장률, 수리률 및 작업전이 시간만을 고려한 클러스터 시스템의 평형 상태 확률 계산. (식 (1)~식 (5) 참조)

Step 2: 고객 도착률(α) 및 시스템 서비스율(β)로부터 $m_0 \ge \lceil \rho \rceil$ 값 계산

Step 3: for $i = m_0, m_0+1, \ldots, n, \max(n,1), \max(n,2), \ldots, \max(n,k)$ do {

Step 4: 1) 대기시간 분포(식 (7) 참조)로부터, 서버의 대수가 i일 때, $W_q(d) = pr\{W \le d\} < \Phi$이면, let i = i+1, $X_i = 0$ and go to Step 3

2) $W_q(d) = pr\{W \le d\} \ge \Phi$이면, 대기시간 데드라인(D^*)에 근거하여 indicator variable X_i 값의 결정: $d \le D^* : X_i = 1$, $d > D^* : X_i = 0$

Step 5: if $X_i = 1$ then put $X_{i+1} = X_{i+2}, \ldots, = X_{\max(n,k)} = 1$ and go to Step 6 else if $X_i = 0$ then let i = i+1 and go to Step 3

}

Step 6: 대기시간 데드라인 성능을 고려한 가용도 A_a의 계산(식 (8) 참조)

3. 실험 및 성능 평가

사용자가 느끼는 현실적인 클러스터 시스템의 가용성능을 구하기 위해, (n,k)-way 클러스터 시스템을 구성하는 주서버와 여분서버의 가동 대수, 요청 대기 시간, acceptance ratio 등 다양한 입력 파라미터 변화에 따른 가용도, downtime 및 손실비용의 관계에 대한 그래프를 작성하였다. 시스템 가용 성능 평가에 사용된 척도에 대한 정의는 다음과 같다.

3.1 downtime

하루 24시간을 기준으로 하여, s 개의 구간으로 나누어진 시간대 별로 도착률(α_i)을 변화시킨 downtime에 대한 정의는 다음과 같다.

$$downtime = \sum_{i=1}^{s}(1-A_{\alpha_i})*(t_i - t_{i-1}) \quad ,t_0 = 0,\ t_s = 24$$

단, A_{α_i} : 시간대 i에서 도착률이 α_i일 때의 가용도

예를 들어, 하루를 3개의 시간대(s=3)로 나누어 심야시간(0~8시), 근무시간(8~16시), 저녁시간(16~24시)으로 구분하고, 각 시간대의 도착률을 각각 α_1, α_2, α_3라고 할 경우 downtime 계산은 아래와 같다.

$$downtime = (1-A_{\alpha_1})*t_1 + (1-A_{\alpha_2})*(t_2 - t_1) + (1-A_{\alpha_3})*(24-t_2)$$

3.2 시스템 파라미터

실험에 사용된 (n,k)-way 클러스터 시스템의 운영 파라미터는 표 1과 같다[1,2].

클러스터 시스템의 가동 일수(T)는 365일(1년)로 하며, 서버의 고장(λ)은 1년에 4회, 고장 수리(μ)에 12시간이 소요된다. 각각의 시간대 별로 시간당 4200, 5400, 4800회씩 작업 요청이 발생하며, 클러스터 서버 한 대는 매 시간당 1200 개의 트랜잭션을 처리할 수 있다고 가정한다. 주서버의 고장으로 인해 발생하는 여분서버로의 작업전이시간($1/\lambda_s$)은 평균 20초가 소요되며, 시스템의 불시 정지로 인해 발생하는 비용(C_i)은 시간대 별로 다른 값을 갖는다. 대기시간 데드라인은 1분이며, 이 조건을 만족하는 비율이 85% 이상일 경우 시스템이 가용하다고 가정한다. 또한 클러스터 시스템의 전반적인 가용성능을 평가하기 위하여, 시간대 별로 계산되어진 가용도로부터, 평균 가용도를 구하였으며(식 (9) 참조), 이를 본 실험 그래프의 가용도 척도로 사용하였다.

$$Mean\_A_{\alpha} = \frac{1}{s}\sum_{i=1}^{s} A_{\alpha_i} \qquad (9)$$

그림 5에서는 부하를 고려하지 않은 상위 단계의 가용도(식 (6) 참조)를 서버의 고장 주기와 가동 대수의 변동에 따라 표시하였다. 주서버와 여분서버의 대수가 증가하고, 고장 주기가 길어질수록 가용도가 높게나왔다. 그렇지만, 여분 서버의 대수가 증가함으로 인해 얻을 수 있는 가용도 증가 효과뿐만 아니라 다수의 서버 설치로 인한 가용도 감소 효과도 있기 때문에, 서버의 신뢰도 목표를 고려하여, 클러스터 시스템의 가동 대수를 결정해야 한다.

그림 6은 대기시간 조건이 고려된 하위 단계의 가용도를 대기시간 조건을 고려하지 않은 상위 단계의 가용도로 나눈 값에 대한 그래프이다. 100%에 가까워질수록 두 계산 방법에 따른 결과 값에 차이가 없다는 것을 의미한다. 여분서버가 없고(k=0), 주서버의 가동대수가 4대 이하인 경우는 가용도 비율이 0%를 나타내고 있는데, 이는 도착률에 따른 대기시간을 고려하여 가용도를 계산할 경우, 대기시간을 일정 수준이하(1분)로 하는 확률을 만족시키지 못하고 있는 것을 나타낸다. 주서버와 여분서버의 대수가 증가할 경우, 두 방법에 따른 계산 결과의 일치 가능성이 점차로 커지는 추세를 나타내고

표 1 시스템 운영 파라미터

연속 가동기간(T)	1 year
서버 고장률(λ)	4 times/year
서버 수리률(μ)	2 times/day
시간대(s)	3 (0~8시, 8~16시, 16~24시)
서비스 요청 도착률(α)	4200, 5400, 4800 transactions/hour
서버 한 대당 서비스률(β)	1200 transactions/hour
작업전이 시간	20 seconds
단위작업 손실비용	1, 10, 5 units/minute
대기시간 데드라인(D^*)	1 minute
Acceptance ratio(Φ)	0.85
Number of stages(r)	20

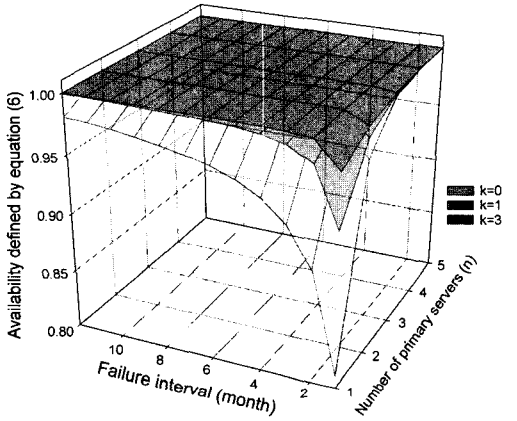

그림 5 The plot of the availability defined by equation (6) according to the number of backup servers(k)

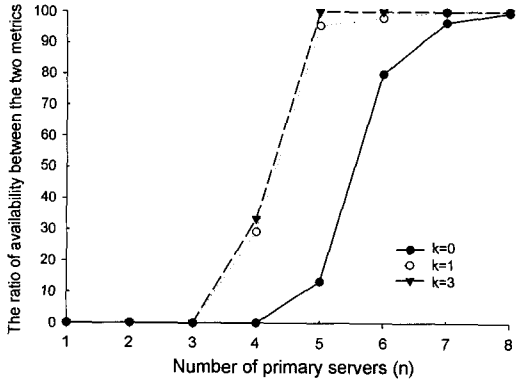

그림 6 The ratio of the proposed availability defined by equation (8) to the availability defined by equation (6)

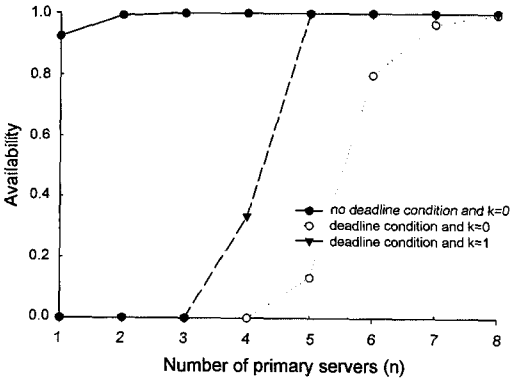

그림 7 The effect of deadline condition to availability

있다. 즉 시스템의 여분용량(spare capacity: 서비스를 처리한 후 남아도는 잉여 용량)이 적을 경우, 현실적인 가용도와 계산상의 가용도에 많은 차이가 발생할 수 있다는 점을 표시하고 있다.

그림 7은 대기시간 데드라인을 고려하지 않는 가용도와 대기시간 데드라인이 고려된 가용도의 변화를 표시한다. 본문에서 언급한 상위단계의 계산을 통할 경우, 시스템의 가용도가 거의 1에 가깝게 나타나고 있지만, 대기시간 데드라인 조건을 고려한 경우, 주서버가 3대 이하일 경우 시스템의 가용도는 0, 즉 발산(대기시간이 무한히 커짐)하는 것을 표시한다. 또한 위 그래프에서 여분서버 없이 주서버의 대수를 증가시키기보다는 주서버의 수가 작더라도 여분서버를 가지는 것이 오히려 가용도 측면에서 유리함을 알 수 있다. 본 연구에서 제시한 가용도 모델에 근거한 실험에서는, (5,0) 시스템 구성보다는 (4,1)의 경우의 가용도가 높게 나타났다. 일반적으로 여분서버의 대수가 증가할수록 가용도가 높아지는 추세를 보였으나, 여분서버 수와 가용도 증가 분은 선형

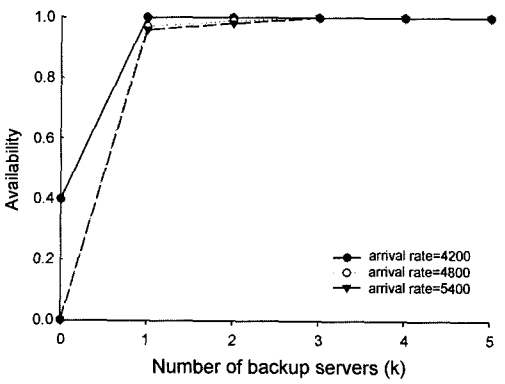

그림 8 The plot of availability according to arrival rates (number of primary servers: n = 5)

관계에 있지 않았으며, 일정 수준이상의 여분서버를 가질 경우, 가용도 증가폭은 상당히 작아지는 것을 알 수 있었다.

그림 8에서는 여분서버와 도착률의 변동에 대한 시스템의 가용도 변화를 파악하고자 하였다. 여분서버가 없는 경우(k=0)와 있는 경우(k=1)의 가용도는 상당히 큰 차이를 나타내고 있으며, 또한 여분서버의 대수가 증가할수록 가용도는 증가하지만 그 상승폭은 미미함을 표시하고 있다. 이는 큐잉 분석 기법의 특성상 주서버가 5대인 경우와 4대인 경우의 이용률의 급격한 변화로 인한 대기시간 증가에 기인하는 것으로 판단된다. 일반적으로 상태 전이도 모델(그림 2 참조)을 통해 시스템의 가용도를 계산할 경우, 대부분의 확률 값이 좌측 초기 상태 부분에 집중하게 된다. (5,1) 시스템의 경우 주서버의 대수가 4대로 되기 위해서는 2대의 서버가 고장이 연속해서 발생해야 하나(초기 상태 부분을 벗어남), (5,0) 시스템의 경우 주서버의 고장으로 인한 (4,0) 상태로의 전이 확률이 비교적으로 높다. 도착률이 작을수록 대기시간이 작아질 확률이 높게됨으로 인해 가용도가 증가하는 것을 알 수 있으며, 이는 본 논문에서 정의한 도착률에 따른 가용도 수식에서 의도하는 바와 일치하는 것을 나타낸다.

그림 9는 대기시간 데드라인(D^*)과 이를 만족시킨 비율(acceptance ratio)의 변동에 대한 가용도 변화를 표시한다. 대기시간 데드라인이 크면 클수록 가용도가 높으며, 만족 비율이 낮으면 낮을수록 가용도는 증가한다. 예를 들면, 대기시간 데드라인이 0.6분(36초 이내의 대기시간)인 그래프는 만족 비율 0.8 근방에서 가용도 저하가 발생한다. 이 경우 D^*를 더 늘리거나, 혹은 만족 비율을 낮출 경우, 원하는 가용도를 확보할 수 있다. 하지만 이때 가용도는 올라가지만 시스템의 성능은 떨어지는 trade-off 가 발생한다.

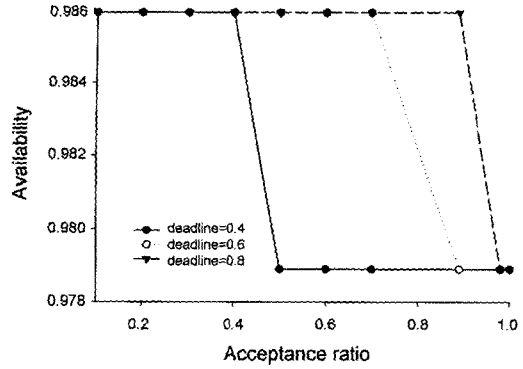

그림 9 The effect of deadline in relation with acceptance ratio (n=5, k=1, λ=4700 trans./ hr.)

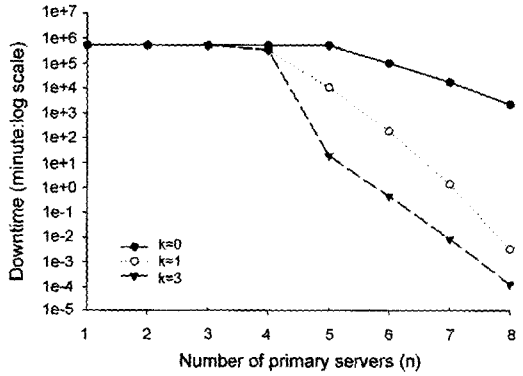

그림 10 The plot of downtime according to the number of backup servers(k)

그림 10은 가용도 저하로 인해 발생하는 시스템의 downtime을 나타낸다. 주서버와 여분서버의 수가 크면 클수록 downtime이 감소하는 추세를 보여주고 있다. 주서버가 4대 이하이거나 여분서버를 두지 않을 경우 downtime이 크게 나왔으며, 그림 7의 가용도 해석과 마찬가지로 (6,0) 시스템보다는 (5,1) 시스템의 downtime 및 손실비용 평가 척도들이 우수함을 나타내고 있다. 일반적인 (n,k)-way 시스템은 시스템의 요구조건(대기시간, 만족 비율)에 따라서 최적 여분서버의 수가 달라질 수 있게 된다. 본 논문의 경우, 원하는 수준의 가용도를 달성하기 위해서, 주서버를 늘리는 것보다는 오히려 여분서버를 최소한 1대 이상 두는 것이 효율적임을 파악하였다.

4. 결 론

본 논문에서는 (n,k)-way 클러스터 시스템의 성능 변화를 고려한 가용도를 계산하기 위하여, 수학적 분석을 통해 구한 상위 단계 closed-form 해와 대기시간 데드라인 성능을 포함하는 하위 단계의 M/M/c 큐잉 기법을 결합하는 새로운 가용도 개념을 제시하였다. 다양한 시스템 운영 상태에 대한 실험을 통해, N-way 클러스터 시스템을 구축할 경우 비록 주서버가 여러 대 동작할 지라도 이들이 동시에 2대 이상 고장날 확률은 매우 작기 때문에, 일정 수준 이상의 수리 능력을 보유할 수만 있다면, 가급적 최소한의 여분서버를 두는 (n,k)-way 클러스터 시스템 구성이 바람직하다는 판단이 가능하였다. 추후에는 1) 여분서버의 여분용량을 결함 감지 및 허용에 사용하는 방법 2) 클러스터 시스템을 구성하는 서버간의 작업 부하 균형 가정을 완화하는 방법 및 3) 결함 발생시 coverage factor를 고려하는 방안 4) 시간대별 서비스률의 변화를 고려하는 방법 등에 대한 연구가 필요하다고 본다.

참 고 문 헌

[1] V. Mainkar, "Availability Analysis of Transaction Processing Systems Based on User-Perceived Performance," Proceedings of the 16th Symposium on Reliable Distributed Systems(SRDS '97), 1997.

[2] 박기진, 김성수, 김재훈, "소프트웨어 재활 기법을 적용한 다중계 시스템의 가용도 분석", 한국정보과학회 논문지(시스템및이론), 제27권, 제8호, pp.730-740, 2000. 8.

[3] R. Buyya, *High Performance Cluster Computing Volume 1: Architectures and Systems*. p. 849, Prentice-Hall, 1999.

[4] 유찬수, "리눅스 클러스터링", 정보과학회지, 제18권, 제2호, pp. 33-39, 2000. 2.

[5] 권세오, 김상식, 김동승, "리눅스 클러스터형 웹 서버 설계", 정보과학회지, 제18권, 제3호, pp. 48-56, 2000. 3.

[6] 오수철, 정상화, "클러스터 시스템 기술 동향", 정보과학회지, 제18권, 제3호, pp. 4-10, 2000. 3.

[7] M.S. Squillante, D.D. Yao and L. Zhang, "Web Traffic Modeling and Web Server Performance Analysis," Proc. of the 38th Conference on Decision & Control, Phoenix, Arizona, pp. 4432-4439, Dec. 1999.

[8] B. Krishnamurthy and C.E. Wills, "Analyzing Factors that Influence End-To-End Web Performance," Computer Networks, Vol. 33, pp. 17-32, 2000.

[9] M. Nabe, M. Murata and H. Miyahara, "Analysis and Modeling of World Wide Web Traffic for Capacity Dimensioning of Internet Access Lines," Performance Evaluation, Vol. 34, pp. 249-271, 1998.

[10] O. Pentakalos, D. Menasce, M. Halem and Y. Yesha, "Analytical Performance Modeling of Hierachical Mass Storage Systems," IEEE Transactions on Computers, Vol. 46, No. 10, pp. 1103-1118, Oct. 1997.

[11] D. Anderson, T. Yang and O.H. Ibarra, "Toward a Scalable Distributed WWW Server on Workstation Clusters," Journal of Parallel and Distributed Computing, Vol. 42, pp. 91-100, 1997.
[12] A.T. Dahbura, K.K. Sabnani and W.J. Hery, "Spare Capacity as a Means of Fault Detection and Diagnosis in Multiprocessor Systems," IEEE Transactions on Computers, Vol. 38, No. 6, pp. 881-891, June 1989.
[13] F. Wang, K. Ramamritham and J.A. Stankovic, "Determining Redundancy Levels for Fault Tolerant Real-Time Systems," IEEE Transactions on Computers, Vol. 44, No. 2, pp. 292-301, Feb. 1995.
[14] B.W. Johnson, *Design and Analysis of Fault-Tolerant Digital Systems,* p. 584, Addison-Wesley Publishing Company, 1989.
[15] H. Nabli and B. Sericola, "Performability Analysis for Degradable Computer Systems," Computers & Mathematics with Applications, Vol. 39, No. 3-4, pp. 217-234, Feb. 2000.
[16] V. Sune, J. Domingo and J. Carrasco, "Numerical Iterative Methods for Markovian Dependability and Performability Models: New Results and a Comparison," Performance Evaluation, Vol. 39, No. 1-4, pp. 99-125, Feb. 2000.
[17] G. Rubino and B. Sericola, "Interval Availability Analysis Using Denumerable Markov Process: Application to Multiprocessor Subject to Breakdowns and Repair," IEEE Transactions on Computers, Vol. 44, No. 2, pp. 286-291, Feb. 1995.
[18] H. Nabli and B. Sericola, "Performability Analysis: A New Algorithm," IEEE Transactions on Computers, Vol. 45, No. 4, pp. 491-494, April. 1996.
[19] B. Sericola, "Availability Analysis of Repairable Computer Systems and Stationarity Detection," IEEE Transactions on Computers, Vol. 48, No. 11, pp. 1166-1172, Nov. 1999.
[20] L. Kleinrock, *Queueing Systems Volume 1: Theory,* p. 417, John Wiley & Sons Inc., 1975.
[21] D. Gross and C.M. Haris, *Fundamentals of Queueing Theory,* p. 556, John Wiley & Sons Inc., 1974.

박 기 진
정보과학회논문지 : 시스템 및 이론
제 30 권 제 6 호 참조

김 성 수
정보과학회논문지 : 시스템 및 이론
제 30 권 제 6 호 참조