

고가용성 Active/Active 클러스터 시스템의 설계

박기진*, 김성수**

*아주대학교 대학원 컴퓨터공학과

**아주대학교 정보통신전문대학원

A Design of Highly Available Active/Active Cluster Systems

Kiejin Park* and Sungsoo Kim**

*Department of Computer Engineering, Graduate School of Ajou University

**Professional Graduate School of Information and Communication Technology, Ajou University

요 약

전자 상거래를 지원하는 클러스터 컴퓨팅 시스템은 장시간에 걸쳐 대량의 트랜잭션을 처리해야 하며, 빈번한 통신 두절 및 데이터 유실 가능성이 높기 때문에, 현재 이와 같은 시스템의 가용도를 개선·확보하기 위한 연구가 활발하다. 본 논문에서는 Active/Active 클러스터 시스템의 소프트웨어적인 결함 발생을 미연에 방지할 수 있는 능동적 결합허용 기법인 소프트웨어 재활(rejuvenation) 방법을 적용하여, 가용도 및 시스템 가동 중단으로 인한 손실비용을 제한하였으며, 이를 통하여 소프트웨어 재활을 하면 할수록, Active/Active 클러스터 시스템의 가용도와 손실비용이 모두 감소하는 추세에 있음을 확인하였다.

1. 서론

일반적으로 컴퓨팅 시스템의 가용도를 높이기 위해서 가동 중인 주(primary)서버의 역할을 대신 수행할 수 있는 여분(backup)서버를 두는 방법이 주로 사용되며, 주서버와 여분서버간의 작업전이(switchover) 방식에 따라 Active/Active 시스템과 Active/Standby 시스템으로 구분하고 있다[1,2]. 위의 두 가지 방법 중에서 주서버와 여분서버가 동시에 가동되는 Active/Active 방식은 주서버의 역할을 대체하는 시간이 매우 짧지만 여러 대의 기계를 동시에 가동해야 하므로, 시스템 운영비용이 많이 들며, 또한 장시간 가동으로 인해 서버들의 고장 발생 가능성이 커지게 된다. 반면에 특정시간에 주서버만 가동되는 Active/Standby 방법의 경우는 Active/Active 방법에 비해서 작업전이 시간은 길지만 그 대신 비용이 적게 들고 시스템의 하드웨어적인 고장 발생 가능성이 상대적으로 낮다.

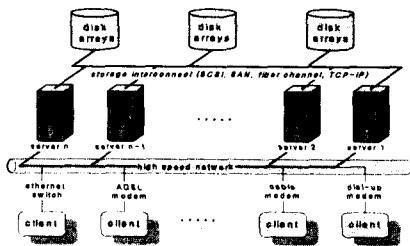


그림 1 Active/Active 클러스터 시스템의 구성

그림 1은 일반적인 Active/Active 클러스터 시스템의 구성을 나타낸다. 클라이언트와 서버는 ADSL(asymmetric digital subscriber line), 케이블 및 랜 등의 고속 가입자망을 통해 연결되며, 서버의 데이터는 SCSI(small computers system interface)나 광 채널 인터페이스 등을 통해 하드 디스크 등의 저장 장치에 저장된다. 고가용성 서비스 시스템 분야(인터넷 뱅킹, 증권, 교환기 등)의 가용도를 확보하기 위해, 여분서

버의 대수를 늘리는 하드웨어적인 결합허용 방법과 하드웨어의 증설이 불필요한 비용 효율적인 소프트웨어 재활(rejuvenation) 방식의 소프트웨어 결합허용 기법이 고려될 수 있으며, 본 논문에서는 고가용도 클러스터 시스템의 결함 발생 사전에 예방할 수 있고, 별도의 하드웨어 자원을 요구하지 않으며, 결합으로 인해 발생하는 손실비용을 최소화 할 수 있는 소프트웨어 재활 결합허용 기법에 관하여 기술하였다.

2. 관련 연구

하드웨어 기술의 발전으로 인해 컴퓨터 하드웨어의 결함 발생률은 상수 값이거나 점차 작아지는 경향이 있는 반면에 하드웨어에 탑재된 소프트웨어의 복잡성 및 크기는 이전에는 상상할 수 없을 정도로 방대해져가고 있기 때문에, 소프트웨어의 결함 발생으로 인한 컴퓨터 시스템의 장애 발생 가능성은 점차 더 높아지고 있다. 클러스터 시스템의 서비스 중단으로 인해 야기되는 가용도 저하 및 손실 비용 발생을 최소화하기 위해, 장시간 시스템 가동으로 인한 소프트웨어 노화[3,4] 현상(memory leak, accumulated round off error, buffer overflow 등)의 진행을 고의로 정지시켜, 클러스터 시스템이 결함 발생 가능성이 적은 건강한 상태에서 재출발하도록 하는 소프트웨어 재활에 의한 결합 예방 방법은 고가용성이 요청되는 Active/Active 클러스터 시스템 응용 분야에서 사용될 가능성이 높다고 볼 수 있다.

본 논문은 이미 발표된 본 저자들의 연구 테마[5]에 대한 계속 연구로써, Active/Active 클러스터 시스템의 가용도 개선을 위해서 소프트웨어적인 결함 발생을 미연에 방지할 수 있는 능동적 결합허용 기법인 소프트웨어 재활 방법에 대한 내용을 포함한다. 구체적으로는, Active/Active 클러스터 시스템의 소프트웨어 재활 과정 및 여분서버로의 작업전이 과정을 semi-Markov 프로세스로 모델링 한 후, 수학적 분석을 통해 구한 시스템의 평형 상태 확률을 이용하여, 다양한 운영 조건하의 가용도 및 손실비용을 계산하였으며, 이를 통하여 소프트웨어 재활을 통한 Active/Active 클러스터 시스템의 가용도 개선 가능성을 확인하였다.

3. Active/Active 클러스터 시스템 모델

소프트웨어 재활을 고려한 Active/Active 가동 방식에 따른 고가용성 클러스터 시스템의 상태 모델을 그림 2에 나타내었으며, 시스템 모델링에 사용된 기본적인 기호들은 다음과 같다.

This work is supported in part by the Ministry of Information & Communication of Korea("Support Project of University Foundation Research<00>" supervised by IITA)

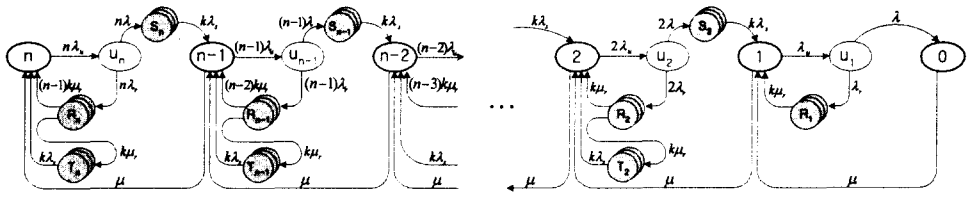


그림 2 Active/Active 클러스터 시스템 상태 전이도(state transition diagram)

- n개의 가동중인 서버로 구성된 Active/Active 클러스터 시스템에서 각 서버의 고장률(λ)은 모든 가동 상태에서 동일하다.
- 고장난 서버의 수리률(μ)은 모든 가동 상태에서 동일하다.
- 서버의 가동을 주기적으로 멈추는 재활률(λ_r)은 모든 가동 상태에서 동일하다.
- 장시간 가동으로 인한 서버의 불안정률(λ_u)은 모든 가동 상태에서 동일하다.
- 재활 작업 시간은 평균값 $1/\mu$, 인 k-stage Erlangian 분포를 따른다.
- 주서버에서 여분서버로의 작업 전이시간은 평균값 $1/\lambda$, 인 k-stage Erlangian 분포를 따른다.
- 재활이 완료된 여분 서버들 중 한 대는 가동중인 주서버의 작업을 전이 받아 주서버 역할을 수행한다.
- 재활 상태와 작업전이 상태를 제외한 모든 상태에서 머무는 시간은 지수분포를 따른다.

정상 상태(normal state)에서 가동되고 있는 서버는 $n, n-1, \dots, 1, 0$ 등의 가동 중인 서버의 수를 상태 변수로 가지고 있으며, 장시간 가동으로 인해 성능이 저하된 불안정 상태(unstable state)의 서버는 $u_n, u_{n-1}, \dots, u_2, u_1$ 로 표시된다. 정상 상태에서 불안정 상태로의 변화율은 $i \cdot \lambda_u$ (i : 가동 중인 서버의 수)로 표시되며, 이는 소프트웨어의 장시간 가동으로 인한 시스템의 불안정률을 반영한다. 불안정 상태에서는 $i \cdot \lambda_u$ 의 변화율로 재활 상태에 들어갈 경우, 소프트웨어 노화 현상으로 인한 일시적 결합 발생 가능성은 대부분 제거될 수 있다. Active/Active 클러스터 시스템의 경우, 특정 시점을 기준으로 모든 서버가 가동되고 있기 때문에 불안정 상태(U_i)에서 고장 발생률 $i \cdot \lambda$ 또한 가동 중인 서버 대수와 비례하도록 모델링 된다.

주서버에 고장이 발생할 경우 여분서버에서 작업을 대체 수행하는 작업전이 상태를 회색의 겹쳐진 원으로 표시된 $S_n, S_{n-1}, \dots, S_3, S_2$ 로 나타냈다. 서버의 대수가 1 대인 단일계(simplex) 시스템에서 서버간 작업전이 과정은 불가능하므로 모델링에서 제외하였다. 또한 회색의 겹쳐진 원모양의 $R_n, R_{n-1}, \dots, R_2, R_1$ 은 재활 상태를 표시하며, 시스템의 가동을 고의로 중지시켜, 시스템이 결합 발생 가능성이 작은 건강한 상태로 되돌리는 것을 나타낸다. 재활이 완료된 여분 서버들 중 한 대가 가동중인 주서버의 작업을 전이 받아 주서버 역할을 수행하는 상태를 $T_n, T_{n-1}, \dots, T_3, T_2$ 로 모델링 하였다. 그림 2의 모델링은 재활 상태와 작업전이 상태에서 머무는 시간(sojourn time)의 분포가 memoryless 성질을 만족하지 않는 k-state Erlangian 분포를 따르기 때문에, semi-Markov 프로세스 문제로 분류되어 정확한 해를 구하는 간단한 방법이 존재하지 않는다. 이를 해결하기 위해서, 그림 2에서 회색으로 칠해진 재활 상태와 작업전이 상태에서 머무는 시간을 임의의 수 k개로 나눈 분할 상태 전이도로 모델링 하였다(그림 3 참조). 예를 들면, 임의의 작업전이 상태 S_i 는 k 개의 분할 상태를 가지는 부분 작업전이 상태(sub-switchover state: S_{ij})들로 표현되며, 각 부분 작업전이 상태에서의 작업전이 소요 시간은 평균 $1/k$ 인 지수분포를 따른다. 그림 3의 분할 상태 전이도는 k 개의 지수분포의 합, 즉 k-stage Erlangian 분포로 표현되며, k-stage Erlangian 분포의 성질 중 상태 분할 개수(k)가 증가함에 따라 확률 분포의 분산 값이 점점 작아져서 결국에는 1의 확률로 unit impulse 값을 가지는 사실을 적용하여, 분

모델링에서 요구되는 재활 상태와 작업전이 상태에서의 확정 값을 가지는 sojourn time 문제를 해결하였다.

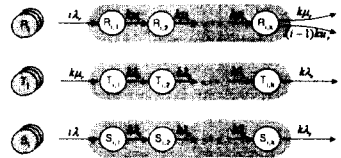


그림 3 sojourn time의 분할 상태 전이도

그림 2의 균형 방정식(balance equation)과 각 상태에서 머물 확률의 총합이 1 이 되는 conservation 방정식을 결합한 연립 방정식을 풀면, 시스템이 평형일 때, 각 상태에서 머물 확률을 얻을 수 있다.

• 이중계 시스템 이상일 때($n \geq 2$)

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

• 단일계 시스템일 때($n=1$)

$$P_1 = \left[1 + \frac{\lambda_u}{\lambda + \lambda_u} \left(1 + \frac{\lambda_r}{k\mu} \right) + \frac{(k-1)\lambda_u}{k\mu} \frac{\lambda_u}{\lambda + \lambda_u} + \frac{\lambda_u}{\lambda + \lambda_u} \frac{\lambda_u}{\lambda + \lambda_u} \right]^{-1} = \left[1 + \frac{\lambda_u}{\lambda + \lambda_u} \left(1 + \frac{\lambda_r}{\mu} + \frac{\lambda_u}{\lambda + \lambda_u} \right) \right]^{-1}$$

$$P_i = \frac{n!}{i!} \left(\frac{\lambda_u}{\lambda + \lambda_u} \right)^{n-i} P_n, \quad i=0,1,2,\dots,n$$

$$P_{u_i} = \frac{\lambda_u}{\lambda + \lambda_u} P_i, \quad i=1,2,\dots,n$$

$$P_{R_{ij}} = \frac{i\lambda_r}{k\mu} \frac{\lambda_u}{\lambda + \lambda_u} P_i, \quad i=1,2,\dots,n, j=1,2,\dots,k-1$$

$$P_{R_{i1}} = \frac{\lambda_r}{k\mu} \frac{\lambda_u}{\lambda + \lambda_u} P_i, \quad i=1,2,\dots,n$$

$$P_{S_{ij}} = \frac{i\lambda}{k\lambda + \lambda_r} \frac{\lambda_u}{\lambda + \lambda_u} P_i, \quad i=2,3,\dots,n, j=1,2,\dots,k$$

$$P_{S_{i1}} = \frac{\lambda_r}{k\lambda + \lambda_r} \frac{\lambda_u}{\lambda + \lambda_u} P_i, \quad i=2,3,\dots,n, j=1,2,\dots,k$$

4. 실험 및 성능 평가

① 가용도

소프트웨어 재활 기법을 적용한 Active/Active 클러스터 시스템의 가용도(Avail)는 모든 서버의 동작이 멈춘 고장 상태(P_0), 시스템 서버스가 일시적으로 중지되는 단일계 시스템의 재활 상태 및 모든 작업전이 상태에서 머물 확률을 제외함으로써 계산된다.

$$Avail = 1 - \left(P_0 + \sum_{i=1}^n P_{R_{ij}} + \sum_{i=1}^n \sum_{j=1}^k P_{S_{ij}} + P_{S_{i1}} \right)$$

② 손실 비용

클러스터 시스템의 갑작스런 가동 정지로 인한 단위 시간당 손실 비용을 C_f , 재할 작업으로 인한 단위 시간당 손실 비용을 C_r , 작업전이로 인한 단위 시간당 손실 비용을 C_s 라 할 경우, 일반적으로 예상 가능한 시스템 정지 비용은 복시 정지로 인한 손실 비용에 비해 훨씬 저렴하게 들므로, $C_f \gg C_r$ 의 관계가 성립하며, 작업전이 비용은 C_f 보다는 작다고 가정한다. Active/Active 클러스터 시스템의 가동 시간에 대한 손실 비용(Cost)은 서버의 가동시간(T)의 함수로 아래와 같이 정의된다.

$$\text{Cost}(T) = \left[C_f \times P_0 + C_r \times \sum_{j=1}^k P_{R,j} + C_s \times \sum_{i=2}^k \sum_{j=1}^k (P_{r,i,j} + P_{s,i,j}) \right] \times T$$

실험에 사용된 Active/Active 이중계 클러스터 시스템의 운영 파라미터는 표 1과 같다.

표 1 시스템 운영 파라미터[5]

시스템 운영 방식(n)	2 (이중계)
연속 가동기간(T)	1년
서버 고장률(λ)	1회/년
서버 수리률(μ)	2회/일
서버 불안정률(λ_u)	1회/15일
재할률(λ_r)	1회/월
재할작업 시간($1/\mu_r$)	10분
작업전이 시간($1/\lambda_s$)	3분
downtime 비용(C_f)	1000
rejuvenation 비용(C_r)	1
작업전이 비용(C_s)	100
분할 stage 수 (k)	20

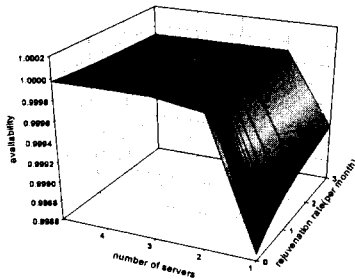


그림 4 재할률과 서버 수에 대한 가용도 변화

그림 4는 Active/Active 클러스터 시스템에서 소프트웨어 재할을 하지 않는 경우(재할률 0)와 재할 실시 주기를 10일 간격으로 약 1년까지 (360일) 변화시키면서, 동시에 시스템을 구성하는 서버의 대수를 단일 계에서 다중계(n=5)까지로 증가시켰을 때, 시스템의 가용도 변화 추세를 나타낸다. 재할을 자주 한다는 의미는 시스템의 불안정성을 사전에 제거하는 것을 의미하며, 이중계 이상의 Active/Active 시스템에서는 재할 작업으로 인한 서비스 중단이 발생하지 않기 때문에, 가용도가 재할률에 비해하는 추세, 즉 재할을 자주 할수록 가용도가 증가하는 현상을 나타냈다. 한편 한 대의 서버만이 가동되는 단일계 시스템의 가용도에 비해서 이중계 이상의 고가용성 클러스터 시스템의 가용도가 높게 나오는 것을 확인할 수 있으며, 이중계 이상의 서버 구성을 통해 개선될 수 있는 가용도의 상승폭은 미미함을 그래프에서 확인할 수 있다. 이는 일반적으로 고가용도를 확보하기 위해 이중계 클러스터 시스템 구성이 가장 비용 효율적임을 나타낸다.

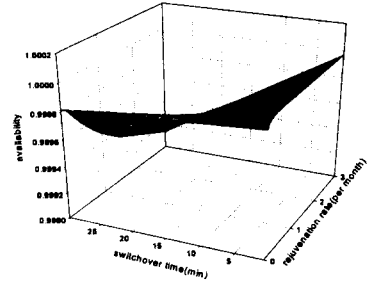


그림 5 재할률과 작업전이 시간에 대한 가용도 변화

그림 5는 작업전이 시간과 재할률이 클러스터 시스템에 미치는 영향을 나타낸다. 작업전이 시간을 0초에서 30분까지 변화시켰을 때, 가용도의 변화는 재할률과 밀접한 관련이 있음을 보이고 있다. 즉 클러스터 시스템을 소프트웨어 재할 정책을 사용하여 운영할 경우, 작업전이 시간 약 15분을 기준으로, 작업전이 시간이 15분 이내일 경우 재할을 하면 할수록 가용도가 높아지는 반면에 작업전이 시간이 15분 이상인 경우, 재할을 하면 할수록 가용도가 감소하는 것을 알 수 있다. 작업전이 시간이 짧으면, 시스템에 하드웨어적 고장 혹은 소프트웨어적인 재할 작업이 발생하여도, 바로 서비스가 가능하게 되므로, 소프트웨어 재할의 효과는 작업전이 시간과 밀접한 관련이 있다. 작업전이 시간이 평균 10분이 소요되고 작업전이 상태의 분할 개수가 20일 때, 각 부분 작업전이 상태의 평균작업전이 시간은 30초가 된다. 20개의 부분작업전이 시간을 모두 합한 20-stage Erlangian 분포의 평균은 지수 분포(k=1)와 동일한 10분이 되지만, 분산값은 지수 분포일 때(100분)의 5%인 5분으로 줄게 된다. 만약 k를 더욱 증가시킬 경우, k-stage Erlangian 분포의 분산값이 k의 비율로 줄게되어 원하는 확정시간을 가지는 작업전이 시간을 표현할 수 있게 된다.

5. 결론

본 연구에서는 Active/Active 클러스터 시스템의 가용도를 모델링하기 위해서, method-of-stages 개념을 적용하였다. 즉 재할 상태와 작업전이 상태에서 머무는 시간을 지수분포들의 합으로 표현한 k-stage Erlangian 분포를 사용하고, 각 상태 분할 개수를 증가시킴으로써 재할 작업과 작업전이 시간을 보다 현실 상황에 근접하도록 모델링 하였다. method-of-stages 방법을 통해 구한 Active/Active 클러스터 시스템 재할 모델의 closed-form 해는 다양한 시스템 운영 상태에 대한 실험을 통해 검증하였으며, 소프트웨어 재할을 통한 Active/Active 클러스터 시스템의 가용도 개선 및 손실비용 절감이 가능하다는 것을 확인하였다. 추후에는 소프트웨어 재할 기법을 적용한 시스템의 부하 및 가동능력을 동시에 고려할 수 있는 방안을 연구할 예정이다.

참고문헌

- [1] B. Johnson, *Design and Fault-Tolerant Analysis of Digital Systems*. p. 584, Addison-Wesley Publishing Company, 1989.
- [2] R. Buyya, *High Performance Cluster Computing Volume 1: Architectures and Systems*. p. 849, Prentice-Hall, 1999.
- [3] S. Garg, A. Puliafito, M. Telek and K. Trivedi, "Analysis of Preventive Maintenance in Transactions Based Software Systems," *IEEE Transactions on Computers*, Vol. 47, No. 1, pp. 96-107, Jan. 1998.
- [4] A. Pfening, S. Garg, M. Telek, A. Puliafito and K. Trivedi, "Optimal Rejuvenation for Tolerating Soft Failures," *Performance Evaluation*, Vol. 27 & 28, North-Holland, pp. 491-506, Oct. 1996.
- [5] 박기진, 김성수, 김재훈, "소프트웨어 재할 기법을 적용한 다중계 시스템의 가용도 분석," *한국정보과학회논문지(시스템및이론)*, 제 27권, 제8호, 2000. 8.