

# 분산 시스템의 결함시 재분배 알고리즘의 선정기준을 위한 특성 분석

## Analysis of Criteria for Selecting Load Redistribution Algorithm for Fault-Tolerant Distributed System

최 병 갑\*, 이 천 희\*\*

Byung-Kab Choi, Cheon-Hee Yi

### Abstract

In this paper, a criteria for selecting an appropriate load redistribution algorithm is devised so that a fault-tolerance distributed system can operate at its optimal efficiency. To present the guideline for selecting redistributing algorithms, simulation models of fault-tolerant system including redistribution algorithms are developed using SLAM II. The job arrival rate, service rate, failure and repair rate of nodes, and communication delay time due to load migration are used as parameters of simulation. The result of simulation shows that the job arrival rate and the failure rate of nodes are not deciding factors in affecting the relative efficiency of algorithms. Algorithm B shows relatively a consistent performance under various environments, although its performance is between those of other algorithms. If the communication delay time is longer than average job processing time, the performance of algorithm B is better than others. If the repair rate is relatively small or communication delay time is longer than service time, algorithm A leads to good performance. But in opposite environments, algorithm C is superior to other algorithms.

## I. 서론

신뢰성 높은 컴퓨터 시스템의 구축을 위한 연구는 초창기부터 끊임없이 요구되었으며 그 결과 다양한 결함허용 기법들이 제시되어 왔다.<sup>(1-5)</sup> 최근에는 분산 시스템의 구축이 일반화되고 있으며 이는 결함허용 기법을 보다 활성화시키는 기반이 되었다.<sup>7)</sup> 결함을 허용하기 위한 다양한 방법중의 한 분야로서 분산 시스템의 노드에 결함이 발생

했을때 결함노드의 부하 재분배를 통한 결함허용 방법도 제시되고 있다.<sup>9)</sup> 그러나 이론적인 알고리즘의 설계에 못지않게 여러가지 알고리즘들중에서 언제 어떠한 알고리즘을 선택하여 사용하여야 하는가라는 실질적인 지침 역시 현실적으로 중요하다. 또한 이러한 지침은 어떠한 상황에서 알고리즘의 선정을 위해 시뮬레이션을 수행하여야 하는지를 알려줄 수도 있다.

본 논문에서는 분산 시스템의 노드 결함시 결함허용을

\* 목원대학교 컴퓨터공학과

\*\* 청주대학교 전자공학과

위한 부하 재분배 알고리즘들을 설계하고 이들의 효율성에 영향을 주는 매개변수들을 변화시켜 일련의 시뮬레이션을 통하여 분석.평가함으로써 특정 환경에서 재분배 알고리즘을 선정하는데 필요한 지침을 제공하기 위한 선정 기준을 제시하고자 한다. 시뮬레이션을 위한 매개변수들로서 작업 도착율, 서비스율, 노드의 고장율과 수리율, 결합노드의 부하 이주에 따른 통신지연 등을 설정하였다. 2장에서는 결합허용을 위한 부하 재분배 알고리즘을 설계한다. 3장에서는 재분배 알고리즘들에 대한 시뮬레이션 프로그램을 작성하고 4장에서는 시뮬레이션의 결과를 분석하여 특정 시스템에서 어떠한 재분배 알고리즘들을 채택할 것인가에 관한 지침을 제시한다. 끝으로 5장에서는 본 연구의 시사점과 향후과제를 논의하였다.

## II. 결합허용을 위한 부하 재분배 알고리즘

본 장에서는 분산 실시간 시스템에서 결합 노드의 부하 재분배 알고리즘을 설정하고 알고리즘을 모델링한다. 분산 시스템에 작업들이 도착하면 정상 상태일 경우 부하 균형 알고리즘이 부하균형을 위하여 여러 노드로 균배하며 결합의 탐지는 결합진단/복구 모듈이 하는것으로 가정한다.

### 2.1 負荷 再分配 시스템의 모델링

분산처리 시스템이 결합일때 부하 재분배 문제에 대한 연구는 2가지 관점으로 부터 접근할 수 있다. 첫번째는 프로세서가 결합일때 부하 재분배 알고리즘의 실행이 시스템의 안정성에 어떻게 영향을 미치는가에 관한 연구이다. 둘째는 프로세서가 결합일때 부하 재분배 알고리즘의 실행으로 지원 프로세서들의 큐의 길이와 큐에서 평균 대기 시간에 어떻게 영향을 미치는가에 관한 연구이다.

본 논문에서는 노드에 결합이 발생했을때 부하의 재분배를 통한 Fail-Soft를 지원하는 n개의 노드로 구성된 분산 시스템으로 가정하였다. 재분배 알고리즘의 시뮬레이션 모델링에 필요한 추가 가정들은 다음과 같다.

[가정 1]시스템 내 각 노드는 새로이 도착하는 작업들을 버퍼링할 수 있는 일 정한 크기의 큐를 갖고, 통신 서브 시스템에는 작업의 이주를 위하여 이주작업 큐 를 갖는다고 가정한다.

[가정 2]결합 노드의 갯수는 임의개 발생할 수있다.

[가정 3]작업의 도착율( $\lambda$ ), 결합 발생율, 작업 서비스율, 고장 수리율은 임의의 확률분포이며, 각 노드의 서비스율은 동일하다.

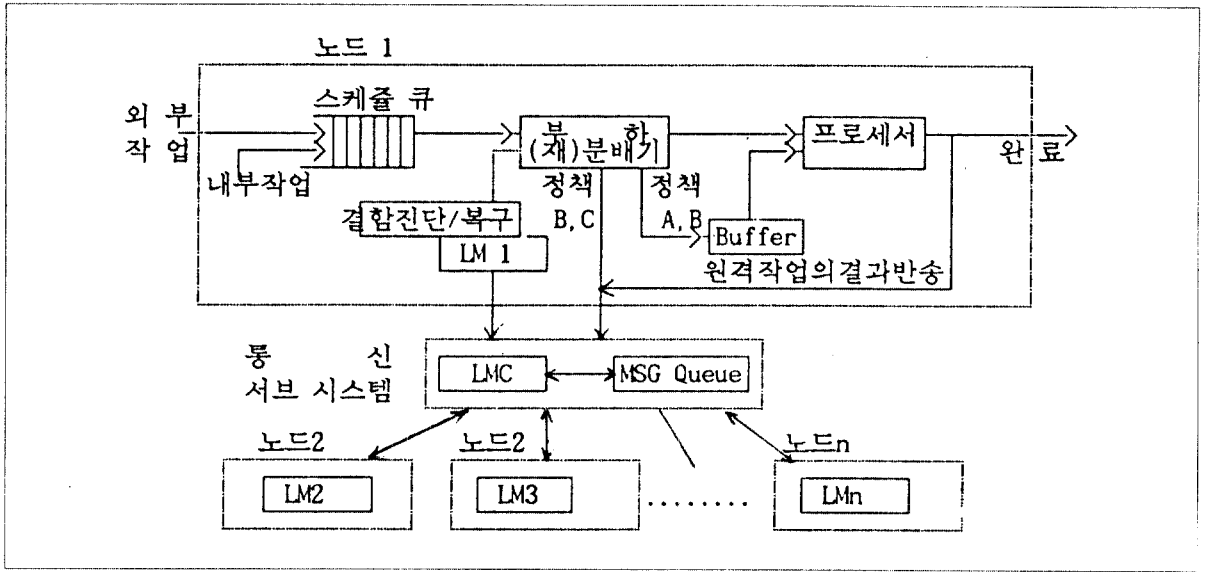
[가정 4]통신 서브시스템의 통신 전용처리기는 각 노드의 대기큐의 상태정보를 관리한다. 부하이주시 노드간의 통신 지연시간(CDT:Communication Delay Time)은 평균 작업 서비스 시간의 배수로 하고 통신결합은 없다고 가정한다.

[가정 5]시스템의 성능 측정지표는 작업 대기 큐와 이주작업 큐 의 평균 길이 로 한다.

어떤 노드에서 결합 발생시 정상으로 작동하는 노드들만의 집합이 재구성되며, 결합 노드의 부하는 정상 노드들간에 균형을 이루도록 재분배하여 실행한다. 결합 노드가 복구되면 정상 노드의 집합을 재구성하되 일단 재분배된 부하는 원상 복구시키지 않는 것으로 가정한다. 각 노드의 큐에는 원격 노드로 부터 부하 분산을 위해 이주되어온 외부작업과 노드에 새로 입력되는 로컬 작업인 내부작업이 대기하게 된다. 결합진단/복구 모듈은 결합의 탐지 사실을 부하 재분배기와 통신 서브시스템의 LMC(Load Management Coordinator)에게 결합 사실을 통보한다. 그러면 LMC는 정상 노드들의 LM(Load Monitor)에게 결합 사실을 통보(Broadcast)하고 타이머를 작동 시킨다. 방송 메시지를 수신한 정상 노드들은 자신이 저부하 상태인 경우(자신의 최대 부하용량과 현재의 부하량의 비교) 자신의 큐에 대기하는 부하량을 반송한다. LMC는 타이머가 작동하는 동안 메시지 큐에 들어온 정상 노드들로부터 가장 작은 부하량을 갖는 노드의 순서대로 결합 노드의 부하를 이주작업 큐(migration Queue)를 통해 재분배한다. <그림 1>은 부하 재분배 시스템의 논리도이다.

### 2.2 부하 재분배 알고리즘의 설계

결합노드의 부하 재분배로 인하여 특정 노드가 과부하되지 않도록 균등화 되어야 한다. 그런데 노드 결합시 대처할 수 있는 방법으로는 노드가 수리.복구될때까지 결합 노드의 큐에서 작업들을 대기시키는 방법, 다른 정상 노드들에게 재분배하는 방법, 끝으로 예비 노드로 이주하여 작업을 계속 실행하는 방법이 사용될 수 있다. 그러나 마지막 방법은 과다한 비용이 소요되므로 본 논문에서는 고



(그림 1) 부하 재분배 시스템의 논리도

려 대상에서 제외하였다. 첫번째와 두번째 방법은 다음과 같이 혼합하여 사용될 수 있다.

수 있을 것이다.

[알고리즘 B] 대기 큐내 부하의 부분적인 재분배를 한다:

결합이 발생한 노드의 대기 큐에 있는 작업 중의 일부 (또는 모두)만을 정상 노드들에게 균배하고 나머지와 수리기간에 도착하는 작업들은 버퍼링 하였다가 수리가 완료된 후 재시작 한다. 이 알고리즘은 단시간 내에 복구가 가능 하거나 통신 비용이 무시 못할 정도인 경우, 또는 치명적인 결과를 초래하지 않을 정도의 하드 실시간 분산 시스템에서는 고려해 볼 수 있는 알고리즘이다.

[알고리즘 C] 모든 작업을 재분배한다:

결합 노드의 대기 큐내의 작업은 물론 수리 중에 도착하는 새로운 작업들도 정상 노드로 재분배 한다. 이는 결합의 발생으로 인하여 치명적인 결과를 초래 하거나 복구가 불가능한 경우 통신 지연의 대소에 관계없이 결합 노드에 할당된 모든 부하는 정상 노드들에게 균배 되어야 한다. 또는 작업의 이주에 따른 통신지연이 별문제가 없을 정도로 아주 작고 모든 부하를 이주 하더라도 시스템의 안정에 영향이 없다고 판단되는 상황에 적용할 수 있을 것이다.

알고리즘 B와 알고리즘 C를 사용할 경우 작업 큐에 대기중이던 작업들의 재분배는 다음과 같이 3가지 유형이

		대기 큐에 있는 작업들	
		대기	재분배
새로 도착하는 작업들	큐에서대기	알고리즘 A	알고리즘 B
	재분배	알고리즘 D	알고리즘 C

여기서 알고리즘 D의 경우는 결합전에 대기하던 작업들이 새로 도착하는 작업들보다 처리가 늦어지므로 평균 응답시간이 길어지게 된다. 알고리즘 D는 실시간 시스템에서 바람직하지 못하며, 따라서 본 연구의 대상에서 제외하였다. 나머지 세가지 알고리즘들은 [8], [9]에서 사용한 알고리즘들과 동일하다.

[알고리즘 A] 부하의 재분배를 일체 실시하지 않는다:

결합이 발생한 노드의 대기 큐에 있는 작업은 물론 수리중에 도착하는 모든 작업들도 버퍼링 하였다가 수리가 완료된 후 재시작 한다. 이 알고리즘은 결합허용을 위하여 부하의 재분배를 실시하지 않는 것으로 짧은 시간내에 수리가 가능 하거나 작업의 이동에 통신 지연이 큰 시스템, 또는 소프트 실시간 분산 시스템 등의 경우에 적용할

있을 수 있다.

- (1) 하나의 지원 노드를 임의적으로 선택하여 결합 노드의 부하를 일괄 이주 시킨다.
- (2) 정상노드의 현재 부하상태와 관계없이 결합노드의 부하를 모든 정상 노드들에게 균배한다(이는 동적 부하균형 알고리즘에 의해 모든 노드가 준균형 상태이었을 것으로 가정한 것임).
- (3) 결합 노드의 부하를 정상 노드들의 현재 부하상태를 고려하여 재분배 후에도 정상 노드들의 부하가 가능한 균형되도록 재분배하여 시스템이 안정 되도록 한다.

(1), (2)는 경우에 따라서는 노드들이 과부하되어 시스템이 불안정 상태가 될 수도 있다. 따라서 본 논문에서는 (3)의 경우만을 고려한다. 이때 각 노드가 균형을 이루도록 구현하는 방법에는 2가지 방법이 있을 수 있다. 첫째는 현재 큐 길이 정보를 사용하여 큐 길이가 가장 짧은 노드로부터 확률적으로 결합 노드의 부하를 재분배 하여 정상 노드들이 균형을 이루도록한다. 두번째 방법은 현재 모든 노드의 수용가능 부하량과 평균 부하량을 구한후 각 정상 노드들의 현재 부하량과의 차이 만큼 노드의 수용가능 부하량의 범위 내에서 결합 노드의 부하를 재분배하여 시스템의 안정성을 취하도록 한다. 이는 시스템의 노드들이 이종으로 구성되어 각 노드의 처리속도와 큐의 길이가 상이한 경우도 적용할 수 있다. 본 논문의 시물레이션에서는 첫번째 방법을 택하였고 균배가 더욱 정확하도록 하기 위하여 한번에 하나의 작업씩 이주하도록하는 방법을 취하였는데, 실제의 시스템에서는 재분배에 상당한 시간이 소요될 수있다.

### Ⅲ. 재분배 알고리즘 선정을 위한 시물레이션 모델

#### 3.1 재분배 알고리즘의 효율성에 영향을 미치는 요인들

부하 재분배 알고리즘의 효율성은 각 노드에서의 작업 큐의 길이에 의해 측정될 수 있다. 큐의 길이에 영향을 주는 환경적 요인으로는 크게 작업의 도착율과 작업의 서비스율로 나눌 수 있다. 그런데 작업의 서비스율은 동일하다고 가정하였기 때문에 본 논문에서는 작업의 도착율

에 관한 요인들을 중심으로 분석한다.

각 노드에 도착하는 작업의 유형은 두가지로 나눌 수 있다. 첫째는 각 노드의 사용자들로 부터 입력되는 내부 작업들이며 둘째는 결합이 발생된 다른 노드로부터 전송된 외부작업들이다. 결합노드로부터 전송되는 작업의 수에 영향을 미치는 요인으로는 결합발생간격시간, 결합수리시간, 통신지연시간 등이다. 결합발생간격시간이란 각 노드에 결합이 얼마나 자주 발생하는가를 의미하며 결합수리시간이란 일단 발생된 결합을 수리하는데 얼마나 오랜 시간이 소요되는가를 의미한다. 통신 지연시간이란 결합노드로부터 하나의 작업을 정상 노드로 분배하는데 소요되는 시간을 의미한다. 그러나 본 논문에서의 통신지연은 부하의 재분배로 인한 이주작업 큐의 길이에 미치는 영향만을 고려하고 정상노드로 재분배할 부하량을 결정하기 위한 정보의 송수신에 필요한 통신지연시간은 고려하지 않았다.

큐의 길이에 영향을 주는 요인들은 재분배 알고리즘의 효율성을 결정하는 요인들이라고 할 수 있다. 따라서 어떠한 환경에서 어떤 재분배 알고리즘이 가장 효율적인가에 관한 결정은 큐의 길이에 영향을 주는 요인들을 고려하여 이루어져야 한다. 결합발생간격시간이 대기행렬의 길이에 미치는 영향은 작업도착율과 유사할 것으로 추정되므로 본 논문에서는 결합발생간격시간이 각각2500, 1500인 경우 작업의 도착율, 결합수리시간 및 통신지연시간을 변화시키면서 시물레이션함으로써 이들이 재분배 알고리즘의 효율성에 미치는 영향을 분석, 평가한다.

#### 3.2 시물레이션 모델

Chou등은 분산 시스템에서 하나의 프로세서가 결합일 때 시스템의 동작을 분석하기 위하여 부하 재분배 알고리즘에 대한 시스템의 안정성 평가를 하였다.<sup>(8)</sup> 연구결과 알고리즘 A는 매우 안정된 알고리즘으로 보일 수 있으나 불안정한 시스템으로 유도될 수 있고 알고리즘 B는 대체로 안정적인 반면 알고리즘 C는 결합노드의 부하량에 좌우된다고 하였다. 그러나 이들의 연구결과를 사용하여 실제의 분산 시스템 환경에서 어떠한 알고리즘을 적용할 것인가를 결정하기는 다음과 같은 문제점들이 내재되었기 때문에 현실적이지 못하다.

- ① 대기행렬의 길이에 관하여 단정적인 예측을 제시하

지 못하고 범위만을 제시했기 때문에 가능한 대기행렬 길이의 범위가 중첩되는 경우 알고리즘의 선택에 어려움이 따른다.

② 재분배 알고리즘들간의 상대적인 효율성 평가가 아니라 각 알고리즘의 추상적인 안정성 평가에만 국한하였기 때문에 분산 시스템의 특정 환경에 따라 어떤 알고리즘을 선택할 것인가의 명확한 기준을 제시하지 못한다.

③ 분산 시스템에서 단지 하나의 노드만 결합이 발생한다고 가정하였고 통신상의 지연시간은 전혀 고려하지 않았다. 그러나 결합은 임의개 발생할 수 있으며 분산 시스템에서 통신의 지연은 다중 프로세서 시스템과는 달리 무시할 수 없다.

본 논문에서는 Chou등의 해석적 연구에 따르는 제약들을 극복하기 위하여 시뮬레이션 접근방법을 사용하였다. 해석적 방법은 복잡한 시스템의 대기행렬 길이에 관하여 개략적인 범위만을 확인할 수 밖에 없는데 반하여 시뮬레이션 방법은 비록 확률적이지만 구체적인 대기행렬 길이의 값을 산출할 수 있기 때문이다.<sup>11)</sup> 본 논문에서는 시뮬레이션을 위하여 SLAM II언어를 사용한다.<sup>12)</sup> SLAM II는 각종 시뮬레이션 연구에 자주 사용되는 비교적 최근에 개발된 전용 시뮬레이션 언어으로써 범용 언어에 비해 모델의 수정과 확장을 용이하게 한다.

<그림 2>는 부하 재분배의 알고리즘C에 대한 SLAM II 네트워크 모델을 도시한 것이다. <그림 2>에서 알고리즘 C는 결합이 발생되었을 때 새로이 발생된 작업들이나 버퍼에 대기하고 있던 작업들은 통신 서브시스템을 통하여 다른 노드들로 재분배된다. 부하의 재분배는 두가지 종류의 시간지연을 발생시킨다. 첫째는 부하균형을 위하여 노드를 탐색하는데 소요되는 시간이며, 둘째는 작업의 전송에 소요되는 시간이다.

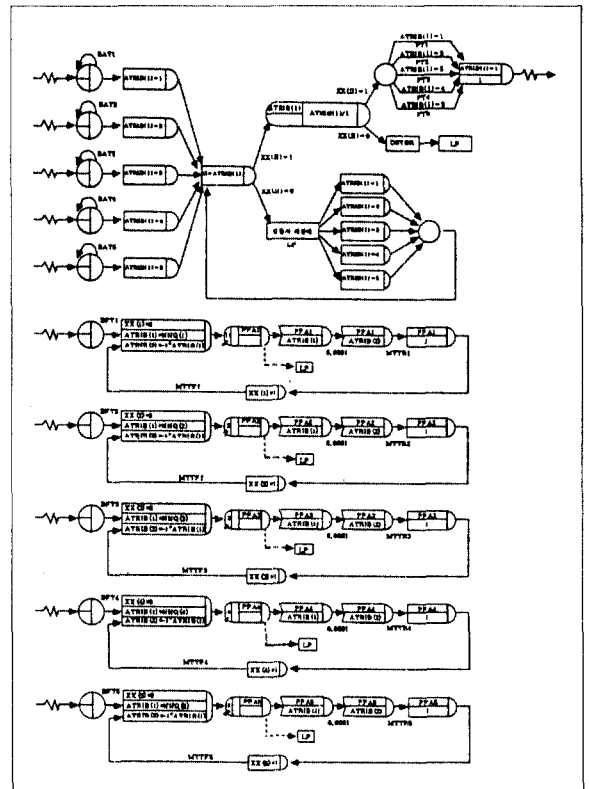
### 3.3 시뮬레이션에서 사용되는 매개변수와 설정치들

작업의 서비스율은 평균 1의 지수분포를 이룬다고 가정한다. 작업의 도착간격시간(1/λ)이 작을수록 작업의 부하는 크다. 만약 작업의 도착간격시간이 작업처리시간보다 작아지면 큐길이는 무한히 증가하게 된다. 그러므로 본 논문에서는 작업의 도착율은 작업처리시간인 1보다 작다고

가정하였다. 평균 결합발생간격시간(MTTF)과 평균 고장수리시간(MTTR)의 설정치들은 CYBER 175와 IBM 360/75 프로세서의 실제 결합율과 수리율의 실측치들을 근사 적용 하였다. 프로세서의 결합율은  $1.4 \cdot 10^{-4}$  faults/min ~  $2.0 \cdot 10^{-3}$  faults/min이고 평균은  $4.0 \cdot 10^{-4}$  faults/min이다. 프로세서의 수리율은 0.2 Repairs/min ~ 0.006 Repairs/min 정도로 알려졌다.<sup>18)</sup> 이들은 모두 지수분포의 형태로 모델에 반영되었다(<표 1> 참조).

<표 1> 시뮬레이션 모델의 매개변수

작업 도착간격시간 (1/λ)	평균고장수리시간 (MTTR:1/β)	통신지연시간 (CDT)	평균결합발생간격시간 (MTTF:1/γ)
1.5	10	0.0	2500
1.4	25	1.0	1500
1.3	50	1.5	
1.2	100	10.0	
1.1	250	20.0	



<그림 2> 알고리즘 C의 SLAM II 네트워크 모델

노드의 고장 수리시간은 25를 기준으로 한다. 25보다 낮은 고장 수리시간과 높은 고장수리시간에 관하여 시물레이션을 수행하여 본다. 통신지연시간은 작업처리시간의 0배, 1배, 1.5배로 상정하여 본다.<sup>(13)</sup> 작업처리시간의 0배라는 가정은 통신 지연시간이 전혀 발생되지 않음을 의미한다. 이 가정은 통신지연시간을 고려하지 않고서 재분배 알고리즘의 안정성을 평가한 기존의 연구들과 비교하기 위한 것이다.

#### IV. 재분배 알고리즘의 선정기준 분석

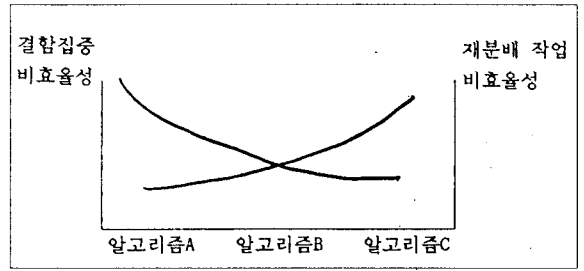
##### 4.1 재분배 알고리즘의 특성

세가지 재분배 알고리즘들의 효율성에 대한 개괄적인 이해를 위하여 재분배알고리즘의 효율성을 결정하는 요인들에 관하여 살펴볼 필요가 있다. 재분배 작업에 관련된 효율성은 두가지 종류의 시간지연에 의해 결정된다.

첫째는 결합의 영향이 특정 노드에 집중됨으로서 발생하는 비효율성이다. 만약 결합이 발생된 노드의 작업들이 정상 노드들로 분배되지 않는다면 그 작업들은 결합수리 시간만큼 기다려야 한다. 비록 다른 정상 노드들은 처리할 작업이 없어서 한가하더라도 특정 노드에 결합의 영향이 집중됨으로 인해서 비효율성이 발생하는 것이다. 결합 집중으로 인한 비효율성은 알고리즘 A와 알고리즘 B에서 발생되지만 알고리즘 C에서는 발생되지 않는다.

둘째는 재분배 작업으로 인한 비효율성이다. 결합이 발생된 노드로부터 대기하던 작업들을 정상 노드들로 재분배시키기 위하여는 최적 노드의 탐색시간과 전송시간이 필요하며 이들 시간만큼 작업처리가 지연된다. 또한 결합 노드의 작업들을 재분배하는 것이 언제나 효율적인 것도 아니다. 작업을 재분배하고난 즉시 결합이 회복되는 경우 결합 노드의 작업 재분배는 오히려 비효율적인 것이다. 이와같은 재분배 작업으로 인한 비효율성은 알고리즘 B와 알고리즘 C에서 발생되지만 알고리즘 A에서는 발생되지 않는다. <그림 3>은 이러한 특성을 보여준다.

특정 환경에서 가장 효율적인 재분배 알고리즘을 결정하기 위하여는 <그림 3>의 재분배 작업으로 인한 비효율성과 결합집중으로 인한 비효율성중 어느 것이 우세한가를 결정하여야 한다. 분산 시스템에서 이러한 비효율성은 평균 대기 큐의 길이(AQL: Average Queue Length)의 증



<그림 3> 재분배 알고리즘 A, B, C의 상대적 비효율성

가로 표현된다. 그러므로 큐의 길이에 영향을 미치는 요인들을 매개변수로 하여 시물레이션하여 봄으로써 어떤 환경에서 어떤 재분배 알고리즘이 적절한지를 판단할 수 있다. 본 논문에서 큐의 길이는 노드의 큐 길이와 부하 재분배를 위한 이주작업 큐의 길이를 고려하였다.

##### 4.2 시물레이션 결과

본 절에서는 앞서 설정한 요인들에 관한 <표 1>의 값을 노드가 5, 7, 10개인 분산 시스템의 환경에 적용시켜 시물레이션을 수행함으로써 각 알고리즘들의 효율성을 측정하였다.

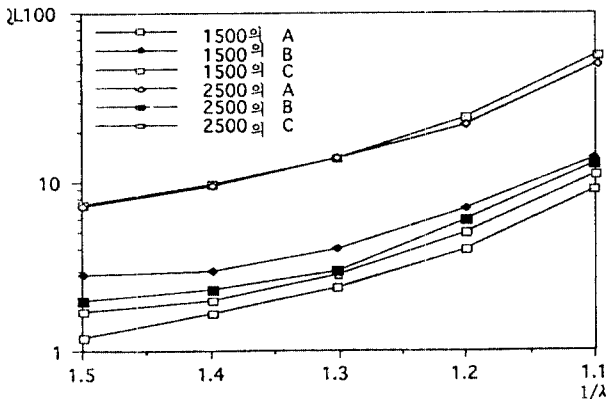
시물레이션 결과에 의하면 작업도착시간간격과 결합발생간격시간, 노드의 수는 알고리즘들간의 효율성 차이에 별다른 영향을 주지 않는 것으로 판명되었다. 물론 이들은 큐의 길이에 영향을 주는 것은 사실이다. 그러나 이들의 변화가 각 알고리즘들에게 상대적으로 미치는 영향의 정도는 유사하다. 그렇기 때문에 작업도착시간간격과 결합발생간격시간, 노드의 수는 어떤 알고리즘을 선정할 것인가에 대해서는 그다지 중요한 기준이라고 할 수 없다.

하나의 예로 <그림 6>으로부터 통신지연이 없고 결합발생간격시간이 2500 이고 결합수리시간이 25인 경우 작업도착시간간격이 알고리즘들에게 미치는 영향을 보면 알고리즘 C가 가장 우수함을 알 수 있다. 알고리즘 B와는 비슷한 큐 길이를 나타내며 알고리즘 A는 C에 비해 4배에서 5배 정도의 큐 길이임을 알 수 있다. 다만 작업도착시간간격이 짧아질 수록 즉 작업 부하가 커질 수록 알고리즘 A와 C간의 효율성 격차는 감소됨을 알 수 있는데 이는 부하량이 커질 수록 알고리즘 C에서의 부하 재분배 작업으로 인한 비효율성이 증가하기 때문인 것으로 해석될

수 있다.

한편 동일 조건에서 통신지연이 1.0인 경우를 보면 알고리즘들간의 효율성에는 변화가 없으나 그 격차가 상당히 감소되었음을 알 수 있다. 이는 <그림 3>에서 보여준 재분배 작업으로 인한 비효율성이 반영되었기 때문이다. 이는 통신지연시간의 영향이며 작업도착시간간격이 알고리즘의 효율성에 미치는 영향은 미미하다는 앞서의 결과를 확인하여 주는 것이다.

고장의 빈도가 상이한 시스템들은 각기 상이한 재분배 알고리즘을 사용해야 하는가는 결합발생간격시간이 재분배 알고리즘의 효율성에 얼마나 영향을 주느냐의 문제이다. <그림 4>는 별다른 영향을 미치지 못함을 보여주고 있다.



<그림 4> 결합간격시간이 큐의 길이에 미치는 영향 (MTTR: 10, CDT: 0)

알고리즘들간의 효율성 차이에 영향을 미치는 주요 요인으로는 통신지연시간, 결합수리시간이다. 통신지연시간이 없거나 평균 작업처리시간과 비슷(1.0 정도)할 경우는 알고리즘 C가 가장 우수했으나 통신지연시간이 그 이상 증가하는 경우는 상대적으로 알고리즘 A와 B는 유리해지나 알고리즘 C는 상대적으로 불리하게 된다. <그림 5(b)>에서 보다시피 알고리즘 B의 효율성이 알고리즘 C보다 우수함을 알 수 있는데 이는 알고리즘 C의 경우 부하 재분배량이 많기 때문이다. 즉 통신지연이 큰 경우는 재분배 작업으로 인한 비효율성이 결합집중으로 인한 비효율성을 훨씬 상회하기 때문이다.

통신지연시간이 작업 큐와 이주작업큐에 미치는 영향은 <그림 5(a)>에 보였다. 통신지연시간이 클수록 이주작업

큐는 알고리즘 C가 크지만 통신지연시간이 서비스 시간과 같거나 작은 경우는 전체 큐의 길이는 알고리즘 B가 크며 수리시간이 길 수록 두 알고리즘들간의 격차는 더욱 커지리라는 것은 쉽게 유추할 수 있다. 이는 알고리즘 B에서 결합수리시간의 증가로 인하여 작업 대기 큐의 길이가 커지기 때문이다. 또한 결합수리시간이 증가하는 경우 결합노드의 작업들을 수리시간동안 대기하도록 하는 알고리즘 A는 상대적으로 불리해 지지만 모든 결합노드의 작업들을 재분배시키는 알고리즘 C는 상대적으로 유리해진다. 이는 결합수리시간이 클 수록 결합집중으로 인한 비효율성이 증가되기 때문이다. <그림 6(a)>로부터 결합수리시간이 클수록 알고리즘 A와 알고리즘 B의 효율성이 기하급수적으로 저하됨을 알 수 있다. 이러한 현상은 통신지연시간을 고려하지 않았기 때문에 알고리즘 C를 과대평가한 것이다. <그림 6(b)>에는 통신지연시간의 변화에 따른 결합수리시간의 영향을 보여준다. 통신지연시간이 증가하는 경우에는, 알고리즘 C가 알고리즘 B보다 저하된 효율성을 보여준다. 이는 앞서 논의한 통신지연시간의 영향이 결합수리시간이 변화될 때에도 나타난다는 점의 의미한다.

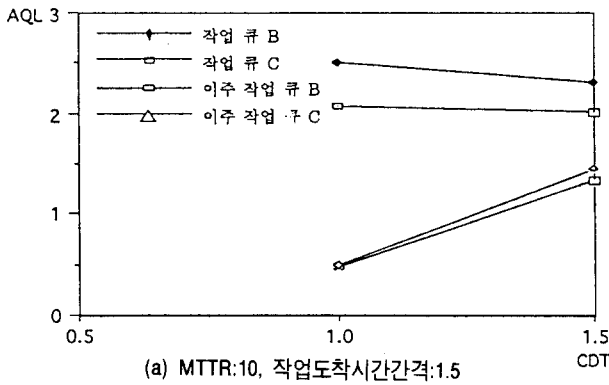
### 4.3 재분배 알고리즘의 선정기준

<그림 4>, <그림 5>, <그림 6>으로부터 알고리즘들간의 효율성에 영향을 미치는 주요 요인들만을 추출하여 요약하면 <표 2>와 같이 정리할 수 있다.

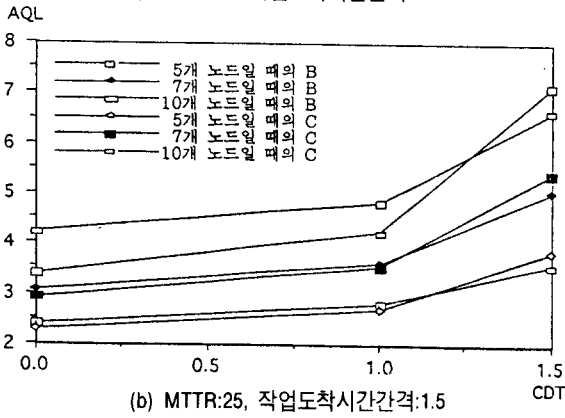
이러한 선정기준은 실제의 분산시스템에서 어떠한 알고리즘을 선정할 것인가를 비교적 명확히 제시한다. 그러나 본 논문에서 시뮬레이션에 사용된 매개변수들중 결합발생간격시간, 결합수리시간, 통신지연시간은 [8], [9]에서 사용한 값들을 상정하여 실행한 결과를 중심으로 분석하여 알고리즘의 선정기준의 지침을 제공하고자 한것이므로 실제 분산시스템 환경에서는 각 매개변수들에 영향을 미치는 요인들에 따른 실제적인 값들에 따라 상이한 결과를 내놓을수도 있기 때문에 보다 정확히 산출하여 적용해야 할 것이다.

시뮬레이션에 사용한 각 매개변수들의 값에 영향을 미칠 수 있는 요인들은 다음과 같이 개략할 수 있다.

- 통신지연시간: 노드간의 거리, 통신선의 종류, 네트워크



(a) MTTR:10, 작업도착시간간격:1.5



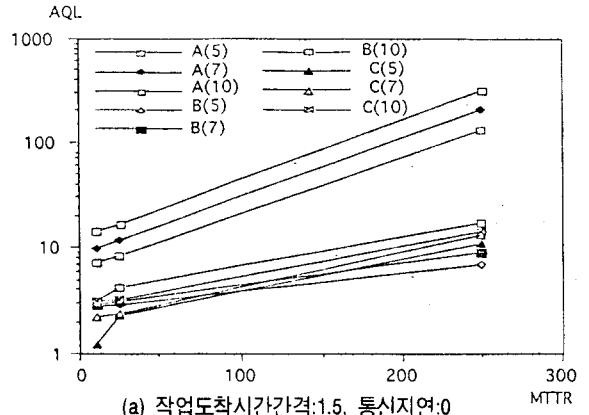
(b) MTTR:25, 작업도착시간간격:1.5

<그림 5> 통신지연이 큐 길이에 미치는 영향

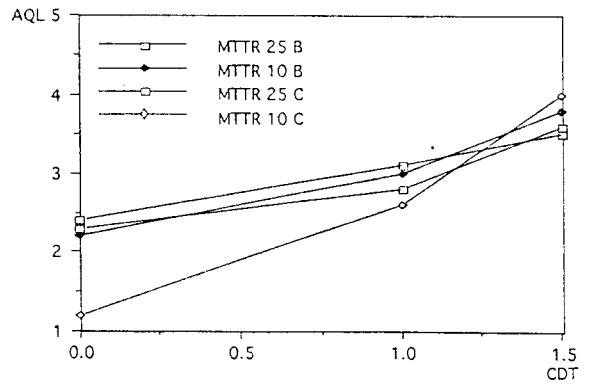
킹 시스템의 토폴로지 등

- 결합수리시간: 시스템 자체의 자동복구 체계의 지원여부, 결합의 내용과 경중, 수리인의 숙련도 등
- 작업도착시간간격: 사용자의 수, 작업의 유형(일괄처리 또는 실시간 처리 등), 다른 노드의 결합 발생을
- 결합발생시간간격: 하드웨어, 소프트웨어, 정보 등의 건장성(Robustness)

또한 이러한 선정기준은 '인제 시물레이션을 수행해야 하는가?'라는 질문에 관한 지침을 제시하기도 한다. 시물레이션을 수행하여야 하는 경우는 위의 표에서 알고리즘들간의 우선순위가 명확하게 결정되지 않은 영역이다. 이 경우에는 구체적인 매개변수를 대입하여 시물레이션을 수행함으로써 세가지 알고리즘들간의 우선순위를 판단하여야 한다.



(a) 작업도착시간간격:1.5, 통신지연:0



(b) 통신지연이 1.0, 1.5일 경우

<그림 6> 결합수리시간이 큐의 길이에 미치는 영향

<표 2> 주요 파라미터와 3알고리즘의 성능간의 관계

		결합수리시간		
		작 다 ( 10 )	중 간 ( 25 )	크 다 (50 이상 - 250)
통신지연 시 간	없다(0.0)	C > B >> A	C > B >> A	C >> B > A
	중간(1.0)	C > B >> A	C > B >> A	C >> B >> A 가능
	크다(1.5)	A >> B > C	A >> B >> C	? 시물레이션 영역

## V. 결론

시물레이션 결과 작업의 도착간격시간, 결합발생간격시간 및 결합수리시간은 재분배 알고리즘들간의 상대적 효율성에 큰 영향을 주지 않았음을 보았다.

통신지연시간과 결합수리시간의 변화는 재분배 알고리즘의 상대적 효율성에 민감한 영향을 주었다. 통신지연시



간이 평균 작업처리시간 보다 큰 경우에는 알고리즘 C보다 알고리즘 B가 우수하며 통신지연시간이 평균작업처리시간과 유사하거나 작을 경우에는 알고리즘 B나 알고리즘 A보다는 알고리즘 C가 현격하게 우수하였다. 매개변수 값들의 변화에 따른 알고리즘의 상대적 효율성이 커다란 변화없이 일반성을 나타낸 알고리즘은 B이었으며 알고리즘 A는 결합수리시간이 작은 경우, 즉 신속한 수리가 가능한 경우와 통신지연시간이 큰 경우에 적합하며 알고리즘 C는 통신지연시간이 작거나 결합수리시간이 큰 경우 적합하였다.

본 논문에서는 시뮬레이션을 통하여 재분배 알고리즘들의 성능을 비교적 구체적으로 평가할 수 있다는 점을 확인할 수 있었다. 그러나 제시된 분석결과가 일반화되기 위하여는 보다 대규모의 분산시스템에 관한 연구가 수행되어야 할 것이다. 본 논문에서 고려하지 못한 한계점과 향후 연구과제는 다음과 같다.

- ① 시뮬레이션 프로그램의 용량한계로 인한 대규모 분산시스템에 관한 시뮬레이션이 어려웠다.
- ② 분산 시스템의 각 노드간 거리 또는 통신속도를 고려하지 않았다.
- ③ 실제 분산 시스템의 환경에서 실행될 수 있는 부하 균형 알고리즘과 노드의 결합시 부하 재분배 알고리즘을 결합한 통합 알고리즘의 구현한다.
- ④ 부하지표가 프로세스 단위인 경우 프로세스간의 통신지연 매개변수를 고려한 재분배 알고리즘을 확장 연구한다.

### 참고문헌

[1] V.P.Nelson and B.D.Carroll, "Tutorial:Fault Tolerant Computing", IEEE Catalog No.EH0254-3, 1987

[2] V.P.Nelson, "Fault-Tolerant Computing:Fundamental Concepts", IEEE Comp. Vo.23, No.7, July, 1990, PP. 19-25

[3] D.A.Rennels, "Fault-Tolerant Computing-Concepts and Examples", IEEE Trans. on Computers, Vol.C-33, No. 12, Dec., 1984, PP.1116-1129

[4] R.K.Iyer, J.H.Patel, W.Kent Fuchs, P.Banerjee, and R. Horst, "HARDWARE and SOFTWARE Fault Tolerance"(ENCYCLOPEDIA of MICROCOMPUTER, Vol. 8), Marcel Dekker, inc., 1991

[5] A.Avizienis, "The N-Version Approach to Fault-Tolerant Software", IEEE Trans. on Software Eng., Vol.SE-11, No.12, Dec. 1985, PP.1491-1501

[6] R.M.Kieckhafer, C.J.Walter, A.M.Finn, P.M.Thambidurai, "The MAFT Architecture for Distributed Fault Tolerance", IEEE Trans. on Computer, Vol.37, No.4, Apr., 1988

[7] J.M.Ayache, J.P.Courtiat, M.Diaz, "REBUS, A Fault-Tolerant Distributed System for Industrial Real-Time Control", IEEE Trans. on Computer, Vol. c-31, No.7, July 1982, PP.637-647

[8] Timothy C.K.Chou, Jacob A.Abraham, "Load Redistribution Under Failure in Distributed Systems", IEEE Trans. on Comp. Vol. C-32, No.9, Sept, 1983

[9] J.S.Park, "A Stability Criterion and Load Redistribution in a Fault Tolerant System", Ph.D.Thesis, Chung Ang Univ., Dec.1990

[10] Law A.M. & W.D.Kelton, "Simulation Modeling and Analysis", McGraw-Hill Inc., 1991.

[11] B.T.Dosh, H.Heffes, "Overload Performance of several Processing Queuing Disciplines for the M/M/1 Queue", IEEE trans. on Comm., Vol.C-34, No.6, June, 1986

[12] A.Alan, B.Pritsker, "Introduction to Simulation and SLAM II", 3rd Ed. Systems Publishing Co., 1986

[13] C.Y Huang, J.W.-S.Liu, "Dynamic Load Balancing Algorithms in Homogeneous Distributed Systems", Proc. 16Th Int. Symp. on Fault-tolerant Computing FTCS-16, PP.216-223, 1986

## ● 저자소개 ●

**최병갑**

1976년 2월 한양대학교 전자공학과졸업  
 1982년 8월 한양대학교 산업대학원 전자계산전공  
 1991년 2월 청주대학교 전자공학과 박사과정 수료  
 1976년2월-1979년7월 삼미종합특수강(주) 전산실  
 1979년8월-1984년2월 한국원자력연구소 선임연구원  
 1984년3월-1986년2월 광주대학교 전산과 전임강사  
 1986년3월-현재 목원대학교 컴퓨터공학과 부교수  
 관심분야: 시스템 소프트웨어, 데이터베이스, 소프트웨어공학 등

**이천희**

1968년 2월 한양대학교 전자공학과 졸업·동대학원졸업  
 1975년 성균관대학교 대학원 전자자료처리학과 졸업  
 1986년 성균관대학교 전자공학과 공학박사학위취득  
 1971년 한국마벨(주) 근무  
 1977년 동양공업 전문대학 전자공학과 근무  
 1979년-현재 청주대학교 전자공학과 교수  
 1983년-1985년 캘리포니아 산호세주립대 객원교수  
 주관심분야: VLSI Layout, ASIC, DRAM, CAD Tool개발 등