

스타형 분산 컴퓨터 시스템의 동적 부하분산을 위한 효율적인 위치결정 정책

임 경 수[†] 하 성 룡^{††} 김 중 곤^{†††}

요 약

스타형의 분산 컴퓨터 시스템에서는 중앙노드의 처리능력이 주변노드들의 처리능력에 비해 클 수가 있고 또 각 노드의 작업도착률은 서로 다를 수 있다. 따라서 작업 전송을 위한 송신자 기준과 수신자 기준의 선택에 따라 부하분산의 성능은 크게 달라진다. 하지만 이종의 분산 컴퓨터 시스템에서 최적 송신자 및 수신자 기준의 설정은 매우 어렵다. 본 논문에서는 동적부하분산을 위하여 고정 임계값 대신에 노드의 평균응답시간이나 작업의 기대응답시간과 같은 동적 기준을 사용하는 효율적인 위치결정 부하 분산 정책을 제안한다.

Effective Destination Decision Policies for Dynamic Load Balancing in Distributed Computer Systems with Star Topology

Kyung Soo Lim[†], Sung Young Ha^{††} and Chong Gun Kim^{†††}

ABSTRACT

In a distributed computer system with star topology, the processing power of central node may be greater than that of peripheral nodes and the job arrival rate of each node may be different. The performance of load balancing may be very different according to the selection of sender criteria and receiver criteria for job transfer. But the optimal decision of sender and receiver criteria is very difficult in heterogeneous distributed computer systems. In this paper we propose effective destination decision policies using dynamic criteria such as mean job response time of a node and estimated response time of a job, instead of fixed threshold for dynamic load balancing.

1. 서 론

분산 컴퓨터 시스템은 컴퓨터 네트워크에 여러 가지 종류의 호스트 컴퓨터(이하 노드라 부름)가 연결되어 하나의 시스템을 이룬다. 이러한 시스템에서는 노드에 따라 과부하의 노드 및 저부하 노드가 존재할 수 있으므로 분산시스템의 자원을 효율적으로 이용하기 위해서는 각 노드간의 부하를 균등히 해야할 필요가 있다. 부하분산은 과부하의 노드로부터 저부하의 노드로 부하를 적

절히 분산시킴으로써 일부 노드에 부하가 집중되는 것을 피하고 작업의 응답시간 단축, 각 노드의 이용률 개선 등 시스템의 성능을 향상시키는데 그 목적이 있다[1,2,3,4,5,6,7,8,9,10].

본 연구에서는 하나의 중앙노드에 각각 하나의 통신링크를 통하여 이기종의 주변노드가 접속된 스타형 분산 컴퓨터 시스템에서의 동적부하분산 문제를 다룬다. 이러한 시스템에 있어 각 노드는 평균작업도착률이나 노드의 처리능력 차이에 의해 과부하 및 저부하 상태가 될 수 있으며 과부하 노드에 도착한 작업을 저부하 상태의 중앙노드 및 주변노드로 적절하게 분산함으로써 시스템의 성능을 향상시킬 수 있다[1,9,10].

† 정 회 원 : 연암공업전문대학 전산계산과 조교수
†† 정 회 원 : 경북전문대학 부교수
††† 정 회 원 : 영남대학교 전산공학과 부교수
논문접수 : 1995년6월17일, 심사완료 : 1995년10월4일

동적부하분산에서 문제가 되는 것은 크게 전송 정책과 위치정책, 정보수집정책들로 구성되는데 [2,3] 전송정책은 각 노드에 도착한 작업을 자노드에서 처리할 지 또는 타노드에서 원격처리할 지를 결정하는 정책이고, 위치정책은 원격처리를 위해 적당한 노드를 결정하는 정책이며, 전송정책이나 위치정책에서 필요한 시스템 정보를 수집하고 유지하는 정책이 정보수집정책이다. 기존의 연구결과는 전송결정 및 목적노드 선택을 위한 시스템 정보의 양에 따라 부하분산의 성능이 차이를 보이고 있으며 정보량이 증가할수록 성능은 개선되나 정보수집에 따르는 오버헤드나 알고리즘의 복잡성 등이 증가된다. 그 중에서도 최소의 시스템 정보만을 이용하는 임계값정책이 상대적으로 간단하면서도 많은 정보를 이용하는 복잡한 방식에 근접하는 성능을 나타내는 것으로 알려져 있다[3,10].

본 연구에서는 기본적으로 송신자 주도형의 부하분산방식을 고려한다. 이러한 부하분산 방식에서 임계값을 이용하는 전송정책은 각 노드에 새로운 작업이 도착될 때마다 각 노드의 작업수가 임계값보다 같거나 클 때 원격처리를 결정하고 임계값보다 적으면 자노드처리를 결정한다. 일반적인 스타형 분산 컴퓨터 시스템에서 중앙노드의 처리능력이 주변노드들의 처리능력에 비해 우수하다는 가정 하에 중앙노드와 주변노드의 송·수신 임계값을 각각 다르게 설정해야만 최적의 성능을 얻을 수 있을 것으로 기대된다[10]. 하지만 다양한 이종 노드 환경에서는 최적의 임계값을 설정하는 것이 용이하지 않음으로 본 논문에서는 노드의 처리능력 및 평균작업도착률 변화에 동적으로 적응할 수 있는 송·수신 기준을 사용하는 새로운 부하분산 방식들을 제안한다. 제안된 방식에서는 작업의 전송 결정을 위한 기준으로 고정 임계값을 이용하거나 각 노드의 평균작업응답시간을 이용한다. 고정 임계값을 사용하는 방식에서는 노드에 새로운 작업이 도착할 때 노드의 보유 작업수와 사전에 결정된 임계값의 비교를 통하여 작업의 전송여부를 결정하고, 평균작업응답시간을 이용하는 방식에서는 새로운 작업을 자노드에서 처리할 때의 기대응답시간과 노드의 평균작업응답시간을 비교하여 작업의 전송

을 결정한다. 한편 송신정책에 의해 전송이 결정된 작업의 전송 목적노드를 찾기 위하여 지역정보만을 이용하여 탐색에 의해 목적노드를 결정하는 방법과 중앙노드의 시스템 전체 상태정보를 이용하여 목적노드를 결정하는 방법을 사용한다. 탐색에 의해 목적노드를 결정하는 방식에서는 탐색 메시지에 전송결정된 작업을 자노드에서 처리할 때의 기대응답시간을 수신기준으로써 포함하여 탐색을 행한다. 탐색이 되는 노드는 탐색 메시지에 포함된 수신기준과 수신되어질 작업을 자신이 처리할 때의 기대응답시간을 비교하여 작업의 수신가능 여부를 결정한다. 중앙노드의 시스템 전체정보를 이용하여 목적노드를 결정하는 방식에서는 작업의 전송을 결정한 노드가 해당 작업의 기대응답시간을 수신기준으로 중앙노드에 제시한다. 중앙노드는 시스템 정보 표의 정보를 이용하여 제시된 수신기준보다 작으면서 해당 작업의 최소기대응답시간을 갖는 노드를 찾고 이를 목적노드의 결정을 요청한 노드에 알린다. 본 연구에서 제안된 방식들은 중앙노드의 주변노드에 대한 상대적인 처리능력의 우수함에 따른 불균형이나, 각 노드의 평균작업도착률의 차이에 따른 불균형에, 적은 비용으로 효율적으로 적응할 수 있음을 컴퓨터 시뮬레이션을 통해 보인다.

본 논문은 2장에서 시스템 모델을 설명하고 제안된 효율적인 부하분산 방식들을 기술하였으며, 3장에서는 실험을 통하여 제안된 부하분산 방식들의 성능을 비교하고, 4장에서 결론을 기술하였다.

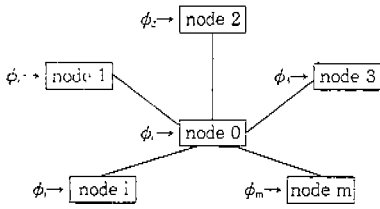
2. 시스템 모델 및 부하분산 정책

2.1 시스템 모델

본 연구에서 대상으로 하는 스타형 분산 컴퓨터 시스템 모델은 (그림 1)과 같다. 모델은 m 개의 주변노드(node 1, node 2, ... node m)들이 하나의 중앙노드(node 0)에 연결된 스타형 분산 컴퓨터 시스템이다.

2.2 부하분산정책

동적부하분산에서는 시스템의 평균 특성에 관한 정보만을 사용하는 정적부하분산[4,8,9]과는



(그림 1) 스타형 분산 컴퓨터 시스템 모델
(Fig. 1) A distributed computer system with star topology

달리 시스템의 순간상태정보에 의존하여 부하를 분산한다[2,3,10]. 동적부하분산은 구현에 있어서 정적부하분산기법에 비해 일반적으로 다소 복잡하나 시스템의 순간상태정보들을 이용함으로써 정적부하분산 때보다 더 나은 성능 향상을 가져올 수 있다.

동적부하분산에는 두 가지 제어형태가 있을 수 있는데, 모든 노드들의 상태정보가 하나의 제어 노드에 모여, 이 정보를 이용하여 제어노드가 작업의 전송을 결정하는 집중형 제어(centralized control)와 작업의 전송결정을 각 노드 자신의 정보를 바탕으로 수행하는 분산형 제어(distributed control)가 있다. 분산형 제어에서는 부하분산의 주체에 따라 송신자주도(sender-initiated)정책과 수신자주도(receiver-initiated)정책 및 혼합(combination)정책으로 나눈다.

본 연구에서는 과부하 노드가 저부하 노드를 찾아 작업을 전송하는 송신자주도정책에 속한 방식으로 목적노드를 선택하기 위하여 탐색을 이용하는 방식과 전체시스템 상태 정보를 이용하는 방식을 고려한다.

다음은 각 방식에서 사용되는 기호와 가절들이다. 여기서 노드번호 0은 중앙노드를 의미한다.

- L_i : 노드 i 의 현재 작업수($i=0,1,2,\dots,m$)
- T_i : 노드 i 의 임계값
- ϕ_i : 노드 i 의 평균작업도착률
- μ_i : 노드 i 의 작업 처리율(처리능력)
- $EJRT_i$: 노드 i 에서의 새로운 작업의 기대응답 시간
($EJRT_i = (L_i + 1) * 1 / \mu_i$)
- $MJRT_i$: 노드 i 의 처리된 작업들의 평균작업응답시간

L_p : 탐색한계

CC : 각 주변노드와 중앙노드간의 작업전송 및 결과의 반환을 위한 평균 시간

스타형 분산 컴퓨터 시스템에서 효율적인 부하 분산을 위한 전송 결정 및 위치결정 방식을 기반으로 하는 새로운 부하분산 방식을 제안한다.

RT(Relative Threshold)

노드 i 에 새로운 작업이 도착할 때 $L_i \geq T_i$ 이면 이 작업이 자노드 처리될 때의 기대응답시간 $EJRT_i$ 를 사용하여 임의의 노드 j 를 탐색하여 $EJRT_i - C < EJRT_j$ ($C = 2 * CC : i \neq j \neq 0, C = CC : i \text{ or } j = 0, i = j$)이면 노드 j 로 작업을 전송하고 그렇지 않으면 L_p 까지 다른 노드를 같은 방법으로 탐색하고 대상 노드를 발견하지 못하면 자노드에서 처리한다.

LBCI(Load Balancing I with Central node)

중앙노드에 작업이 도착하는 경우에는 $L_0 \geq T_0$ 이면 이 작업이 자노드 처리될 때의 기대응답시간 $EJRT_0$ 를 사용하여 시스템 정보 표에서 $EJRT_0 > \min\{EJRT_i + CC \mid i=1,2,\dots,m\}$ 인 노드 i 를 발견하면 주변노드 i 로 작업을 전송하고 그렇지 않으면 작업을 자노드에서 처리한다.

주변노드 i 에 작업이 도착하는 경우에는 $L_i \geq T_i$ 이면 이 작업이 자노드 처리될 때의 기대응답시간 $EJRT_i$ 를 사용하여 중앙노드에 목적노드 결정을 위한 요청을 보낸다. 중앙노드는 요청을 받으면 $EJRT_i > \min\{EJRT_j + C \mid j=0,1,2,\dots,m, j \neq i, C = CC : j=0, C = 2 * CC : j \neq 0\}$ 인 노드 j 를 시스템 정보 표에서 찾아 요청노드에 목적노드 결정을 알리고, 발견하지 못하면 작업을 자노드 처리하도록 요청노드에 응답을 보낸다.

시스템 정보 표에 기록되는 정보는 각 노드의 현재 보유작업수와 노드의 처리능력이다. 이 방식의 정보수집정책은 각 노드의 보유 작업수 변동이 있을 때마다 각 주변노드는 이를 중앙노드에 알려서 항상 현재의 상태 정보를 유지할 수 있게 하는 정책이다[7]. 일반적으로 시스템 정보를 이용하여 목적노드를 결정하는 방식에서의 문

제는 정보수집에 따르는 오버헤드이다. 하지만 정보수집의 오버헤드를 줄이기 위하여 주변노드가 목적노드 결정을 위해 중앙노드에 요청할 때, 중앙노드가 대상 목적노드를 발견한 경우는 시스템 정보 표에서 목적노드로 선정된 노드의 작업수를 1증가 시켜주며 목적노드로 선정된 노드는 타노드의 작업이 전송되어 오면 중앙노드에 작업수의 변동을 알릴 필요가 없다. 또한 중앙노드가 목적노드를 발견하지 못한 경우에는, 시스템 정보표의 요청노드의 작업수를 1증가 시켜주며 요청노드는 작업을 자노드에서 처리하며 중앙노드에 작업수의 변동을 알리지 않는다[7].

LBCII(Load Balancing II with Central node)

이 방식은 한 노드에 새로운 작업이 도착할 때마다 그 작업을 자노드에서 처리할 경우의 기대작업응답시간과 그 노드의 평균작업응답시간(MJRT)을 비교하여 기대작업응답시간(EJRT)이 클 경우 원격처리를 결정하고 기대작업응답시간이 더 작을 경우 자노드 처리를 결정한다. 위치정책과 정보수집정책은 제안된 LBCI방식과 동일하다.

본 연구에서 제안하는 방식과의 성능 비교를 위하여 송·수신 기준으로써 고정 임계값을 사용하는 다음의 FT방식[2,10]을 고려한다.

FT(Fixed Threshold)

송·수신 기준으로 임계값을 사용하고 각 노드는 지역정보인 현재 보유 작업수만을 이용하여 작업의 송·수신을 결정한다.

노드에 작업이 도착할 때 $L_i \geq T_i$ 이면 임의의 노드 j 를 탐색하여 $L_j < T_j$ 이면 노드 j 로 작업을 전송하고 그렇지 않으면 L_p 까지 다른 노드를 탐색하고 적당한 노드를 발견하지 못하면 작업을 자노드에서 처리한다.

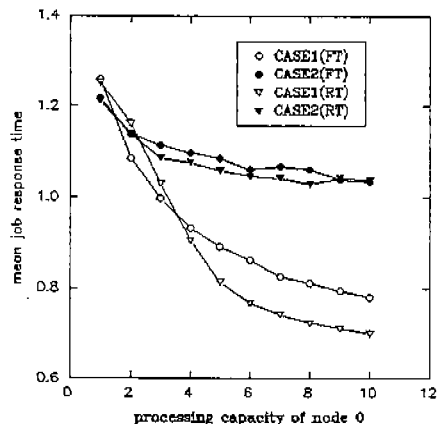
3. 실험 및 결과 분석

SIMSCRIPT II로 IBM PC에서 대상 시스템을 모델링하고 시뮬레이션 하였다. 시뮬레이션에서는 노드 수는 중앙노드를 포함하여 13, 탐색한계

(L_p)를 3, 임계값 T_i 는 1로 두었다. 각 노드의 처리능력 및 각 노드의 평균작업도착률은 각 실험에서 목적에 맞게 설정되었다. 작업의 도착간격 및 서비스 시간분포는 지수분포를 따른다고 가정하였으며 중앙노드와 주변노드간의 작업의 전송 및 결과 반환시간은 평균 0.1의 지수분포를 따른다고 가정하였다. 또한 탐색이나 정보수집에 따른 통신비용은 무시한다. 주요 시뮬레이션의 결과들을 다음에 보인다.

3.1 탐색전략에 따른 성능 비교

(그림 2)는 모든 노드의 평균작업도착률은 0.5, 주변노드들의 처리능력을 1로 둘 때 중앙노드의 처리능력을 증가시키는 경우 탐색에 의해 목적노드를 결정하는 방식에서 탐색 전략에 의한 성능을 비교한다. 중앙노드 처리능력이 주변노드들의 처리능력에 비해 큰 경우 중앙노드를 우선 탐색하는 방식(case 1)이 중앙노드와 주변노드를 구분하지 않고 임의 탐색하는 방식(case 2)보다 부하분산 성능이 우수함을 볼 수 있다. 중앙노드의 처리능력이 주변노드들의 처리능력과 비슷할 때는 중앙노드를 우선 탐색함으로써 중앙노드에게로 작업전송이 집중되어 탐색한계(L_p) 내에서 저부하노드를 찾을 가능성이 상대적으로 줄어든다. 중앙노드의 처리능력이 주변노드들의 처리능력에 비해 충분히 큰 경우에는 중앙노드를

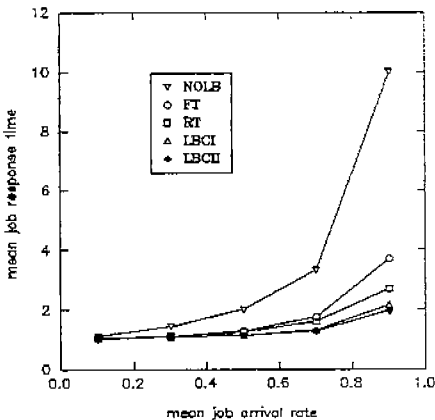


(그림 2) 탐색전략에 따른 성능비교
(Fig. 2) The performance comparison of probing strategies

우선 탐색하여 처리능력이 우수한 중앙노드를 목적노드로 결정하는 것이 작업의 응답시간을 개선할 수 있다. 따라서 이후의 실험들은 모두 중앙노드를 우선 탐색하는 방식을 택한다. 한편 전반적으로 임계값이 아닌 작업의 기대응답시간을 사용하여 목적노드를 결정하는 RT방식이 FT방식보다 우수함을 보이고 있다.

3.2 평균작업도착률에 따른 성능 비교

(그림 3)은 각노드의 처리능력이 1로 동일할 때 각 노드의 평균작업도착률을 동일하게 증가시키는 경우 제안된 방식의 부하분산 성능을 비교한다. 모든 경우 각 방식의 부하분산 성능은 부하분산하지 않는 경우(NOLB)보다 우수하다. 각 노드의 평균작업도착률이 커질수록 기대응답시간을 수신기준으로 사용하는 방식들(RT, LBCI, LBCII)이 고정 임계값을 이용하는 방식(FT)에 비해 부하분산 성능이 우수함을 알 수 있고, 또한 노드의 지역정보만을 사용하는 방식보다는 시스템 전체의 정보를 이용하여 목적노드를 결정하는 방식(LBCI, LBCII)이 더 우수하다. 그 중 노드의 평균작업응답시간을 작업의 송신기준으로 사용하는 방식(LBCII)이 가장 우수하다.

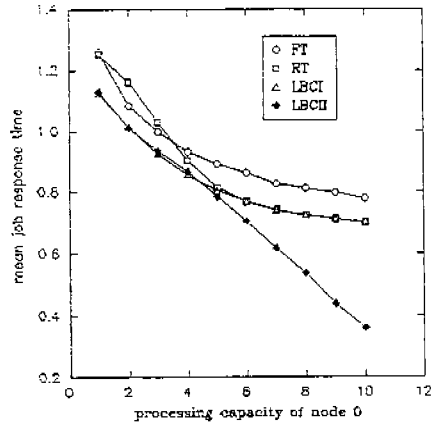


(그림 3) 각 노드의 평균작업도착률에 따른 성능 비교
(Fig. 3) The performance comparison according to mean job arrival rate

3.3 이증 환경에서의 성능 비교

(그림 4)에서는 모든 노드의 평균작업도착률이 0.5, 주변노드들의 처리능력이 1로 동일한 경

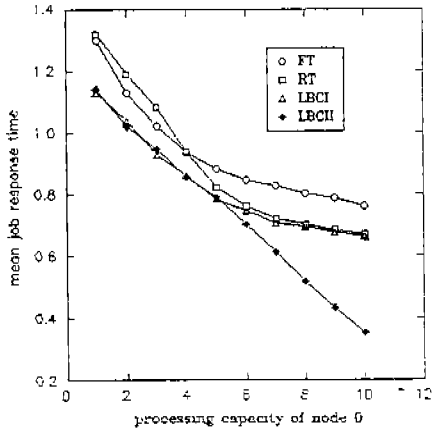
우 중앙노드의 처리능력 증가에 따른 각 방식의 성능을 보인다. 수신기준으로 고정 임계값 대신에 작업의 기대응답시간을 사용하는 방식(RT, LBCI, LBCII)이 중앙노드의 처리능력이 증가할수록 부하분산 성능을 많이 향상시킨다. 탐색을 행하는 방식에서 고정 임계값을 사용하는 FT방식에 비하여 기대응답시간을 수신기준으로 제시하는 RT방식이 중앙노드의 처리능력이 주변노드들의 처리능력에 비해 상대적으로 큰 경우에 FT방식보다 우수하다. 이는 고정 임계값이 중앙노드의 처리능력 증가와 같은 노드의 환경 변화에 능동적으로 적응하지 못하는 기준임을 의미한다. 반면에 작업의 기대응답시간을 이용하여 목적노드를 결정하는 방식은 중앙노드의 성능 증가와 같은 이부하 환경에 잘 적응되어 우수한 부하분산 성능을 보인다. 또한 LBCI방식과 LBCII방식은 전체 시스템 정보를 이용하나 LBCI방식은 작업의 송신결정 기준으로 임계값을 사용하기 때문에 중앙노드의 처리능력 증가와 같은 노드의 환경 변화에 적응하는 최적의 송신 결정이 어렵다. 하지만 LBCII는 각 노드의 평균 작업응답시간을 송신 기준으로 사용함으로써 이를 극복하고 있다.



(그림 4) 중앙노드의 처리능력 증가에 따른 성능비교
(Fig. 4) The performance comparison according to increasing node 0's processing power

(그림 5)에서는 <표 1>의 case 1의 경우로 각 노드의 평균작업도착률이 서로 다른 경우에 중앙노드의 성능증가에 따른 제안된 방식들의 부하분

산 성능을 비교한다. 대체로 부하분산 성능의 경향은 모든 노드의 작업도착률이 동일한 경우의 (그림 4)와 비슷하다.

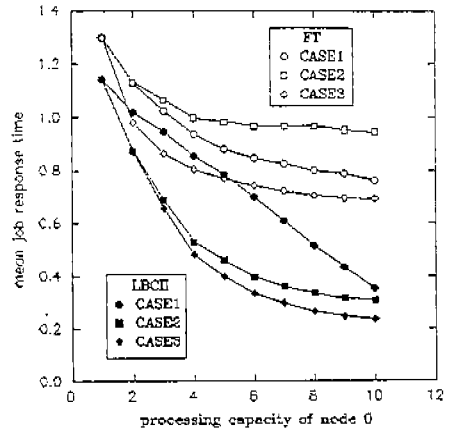


(그림 5) 중앙노드의 처리능력 증가에 따른 성능비교
(Fig. 5) The performance comparison according to increasing node 0's processing power

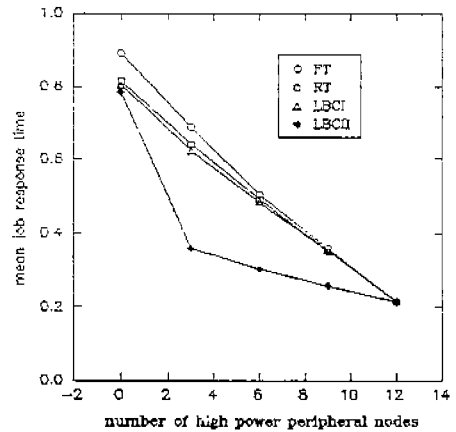
(그림 6)에서는 <표 1>에서와 같이 각 노드의 평균작업도착률이 서로 다를 때 중앙노드의 처리능력을 증가하는 경우(case 1)와 평균작업도착률이 가장 작은 주변노드들의 처리능력을 증가시키는 경우(case 2), 또한 평균작업도착률이 가장 큰 주변노드들의 처리능력을 증가시키는 경우(case 3)에 대해 FT방식과 LBCII방식을 서로 비교한다. 결과에 의하면 어떤 경우라도 FT방식보다 LBCII방식이 성능이 우수하며, 두 방식에서 과부하 노드의 처리능력 증가가 대체로 우수한 성능을 보이고 있다. 한편 스타형 분산 모델에서는 중앙노드의 처리능력이 주변노드들의 처리능력에 비해 월등히 우수한 경우 중앙노드의 처리능력의 증가만으로도 어느 정도 부하분산 효과가 있음을 알 수 있다.

(표 1) (그림 6)에서의 파라메타 값
(Table 1) The parameter values of (Fig. 6)

		node	node	node	node	node
		0	1-3	4-6	7-9	10-12
	ϕ	0.5	0.1	0.3	0.7	0.9
case1	μ	1→10	1	1	1	1
case2	μ	1	1→10	1	1	1
case3	μ	1	1	1	1	1→10



(그림 6) 각 노드의 처리능력 증가에 따른 성능비교
(Fig. 6) The performance comparison according to increasing nodes's processing power

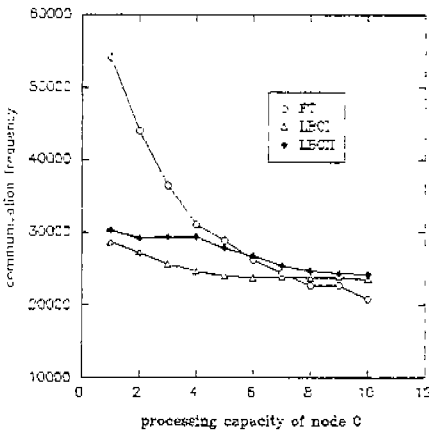


(그림 7) 처리능력이 우수한 주변노드 수에 따른
(Fig. 7) The performance comparison according to the number of high power peripheral nodes

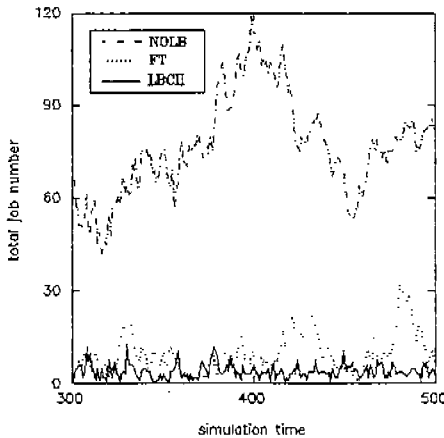
(그림 7)에서는 모든 노드의 평균작업도착률이 0.5이고 중앙노드의 처리능력이 5인 경우에 처리능력이 5인 주변노드들의 수를 증가시킬 때의 각 방식의 성능을 비교한다. LBCII방식이 가장 효율적인 부하분산 성능을 보임을 알 수 있다. 따라서 (그림 6)과 (그림 7)에서 알 수 있듯이 각 노드의 평균작업응답시간과 같은 동적인 수신기준을 사용하고, 또한 작업의 기대응답시간과 같은 동적인 수신기준을 사용하는 방식이 노드들의 처리능력의 변화와 각 노드의 평균작업도착률이 다른 이종 분산 컴퓨터 시스템에 가장 적응력이 있음을 알 수 있다.

3.4 목적노드 결정을 위한 통신회수의 비교

(그림 8)에서는 목적노드 결정을 위한 통신회수의 차이를 비교해 본다. <표 1>의 case 1인 경우에 중앙노드와 주변노드간의 목적노드 결정을 위한 통신회수를 1로 두었을 때 중앙노드의 처리능력 증가에 따른 LBCI, LBCII방식의 정보수집 및 목적노드 선정을 위한 중앙노드로의 요청과 응답에 따른 통신회수와 탐색을 행하는 FT방식에서의 목적노드 탐색 및 응답을 위한 통신회수를 비교한다. 대체로 LBCI과 LBCII방식이 FT방식에 비해 더 적은 통신 비용으로 목적



(그림 8) 목적노드 결정을 위한 통신회수
(Fig. 8) The communication frequencies for destination decision



(그림 9) 부하분산 효과 비교
(Fig. 9) The comparison of load balancing effect

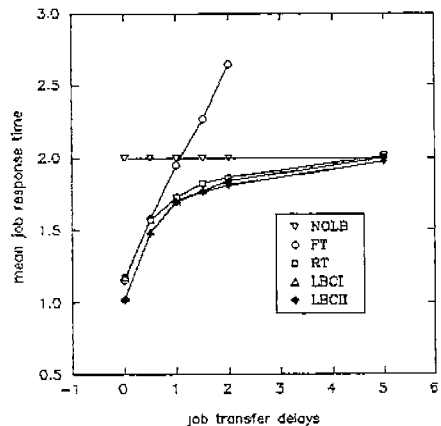
노드를 결정할 수 있음을 알 수 있다. 따라서 스타형의 분산 컴퓨터 시스템에서는 중앙노드에 전체 시스템 정보 표를 유지하는 것이 그다지 큰 비용 없이 우수한 부하분산 성능을 보일 수 있는 방식임을 알 수 있다.

3.5 LBCII방식의 부하분산 효과 비교

(그림 9)는 모든 주변노드들의 처리능력은 1, 중앙노드의 처리능력은 5, 각 노드의 평균작업도착률은 0.5로 동일 할 때 시뮬레이션 시간의 경과에 따라 시스템의 각 노드에 대기중인 작업수를 비교한다. LBCII방식과 FT방식이 부하분산을 하지 않는 경우(NOLB)에 비해 시스템 전체 작업수를 낮게 유지시킴을 볼 수 있으며, LBCII방식이 우수한 부하분산 효과를 보임을 알 수 있다.

3.6 작업전송지연에 따른 성능 비교

(그림 10)은 모든 노드의 처리능력이 1, 작업도착률이 0.5일 때 작업전송지연의 변화에 따른 제안된 방식들의 성능을 비교한다. 단순 임계값과 작업수단의 비교로서 목적노드를 결정하는 FT방식은 작업전송지연이 증가함에 따라 NOLB보다 성능이 나빠진다. 그러므로 이 방식은 작업전송지연의 변화에 따라 새로운 최적 임계값의 설정이 되어야 할 것이다. 반면에 작업전송지연



(그림 10) 작업전송지연에 따른 성능 비교
(Fig. 10) The performance comparison according to job transfer delays

을 포함한 작업의 기대응답시간을 수신 기준으로 사용하는 제안된 방식들(RT, LBCI, LBCII)은 작업의 전송지연 변화에도 적응력이 있음을 알 수 있다. 한편 작업전송지연이 작을수록 LBCI, LBCII방식이 유리함을 알 수 있다.

4. 결 론

본 연구에서는 스타형 분산 컴퓨터 시스템의 동적 부하 분산에서 송신자 주도 전략을 기본으로 각 노드의 처리능력의 차이나 평균작업도착률의 차이에 동적으로 적응할 수 있는 효율적인 부하분산 방식을 제안하였다. 작업의 전송 결정을 위한 기준으로 임계값 및 노드의 평균작업응답시간을 사용하였으며 수신 기준으로 전송 결정된 작업의 기대응답시간을 사용하였다. 탐색에 의해 목적노드를 결정하는 방식에서는 중앙노드의 처리능력이 주변노드의 처리능력에 비해 큰 경우 중앙노드를 우선 탐색하는 방식이 중앙노드와 주변노드를 구분 않고 임의 탐색하는 방식에 비해 성능이 우수하였다. 송신 기준으로 임계값을 사용하는 방식에 비해 노드의 평균응답시간과 같은 동적 기준을 사용하는 방식이, 또한 수신기준으로 작업의 기대응답시간과 같은 동적 수신 기준을 제시하는 방식이 노드의 처리능력의 차이나 작업도착률의 차이, 작업전송지연의 차이에 더 큰 적응력을 보였다. 한편 탐색에 의한 목적노드의 결정 방식에 비하여 중앙노드의 시스템 정보표를 이용하는 방식이 더 우수한 성능을 보였다. 중앙노드에 정보 표를 유지하는 방식에서는 목적노드 결정 요청 및 정보 수집에 따른 통신 비용이 탐색을 행하는 임계값 정책의 탐색에 따른 통신 비용에 비해 그다지 크지 않았다. 따라서 제안된 부하분산 방식 중 LBCII방식이 이종의 스타형 분산 컴퓨터 시스템에서 가장 적응력이 있는 효율적인 부하분산 방식임이 확인되었다.

참 고 문 헌

[1] Anurang Kumar, "Adaptive Load Control of the Central Processor in a Distributed System with a Star Topology",

IEEE Trans. Comp., Vol.38, No. 11, pp. 1502-1512, 1989.

- [2] D. Eager E. Lazowska and J. Zahorjan, "Adaptive Load Sharing in Homogeneous Distributed System", IEEE Trans. Se., Vol. SE-12, No 5, pp. 662-675, 1986.
- [3] D. Eager E. Lazowska and J. Zahorjan, "Comparison of Receiver initiated and Sender initiated Adaptive Load Sharing", Performance Evaluation 6, pp. 53-68, 1986.
- [4] N. tantawi and Don Towsley, "A General Model for Optimal Static Load Balancing in Star Network Configurations", Performance '84, pp. 227-291, 1979.
- [5] R. mirchandaney, Don Towsley and John A. stankovic, "Analysis of the Effects of Delays on Load Sharing", IEEE Trans. Comp., Vol. 38, No 11, pp. 1513-1525, 1989.
- [6] Yuan-Chieh Chieh and Walter H. Kohler, "Models for Dynamic Load Balancing in a Heterogeneous Multiple Processor System", IEEE Trans. Comp., Vol c-28, No. 5, pp. 354-361, 1979.
- [7] Awa-Chun Lin and C.S. Raghavendra, "A Dynamic Load-Balancing Policy with a Central Job Dispatcher (LBC)", IEEE Trans. SE., Vol.18, No.2, pp. 148-158, 1992.
- [8] C-G.Kim and H. Kameda, "An Algorithm for Optimal Static Load Balancing in Distributed Computer Systems", IEEE Trans. Comp., Vol.41, No. 3, pp. 381-384, March 1992.
- [9] 임경수, 김수정, 김종근, "스타형 컴퓨터 네트워크의 부하균형방향 정책", 정보처리논문지, 제1권, 제4호, pp. 427-437, 1994.
- [10] 임경수, 박미희, 김종근, "스타형 분산 컴퓨터 시스템의 동적 부하분산", 영남대학

교 공업기술연구소논문집, 제21권, 제1호,
pp. 89-96, 1993.



임 경 수

1984년 경북대학교 공과대학
전자공학과 (학사)
1986년 영남대학교 공과대학 전
자과 전자계산기전공(공학석
사)
1993년 영남대학교 공과대학 전
자과 전자계산기전공 박사과

정 수료

1991년~현재 연암공업전문대학 전자계산과 조교수
관심분야: 운영체제, 분산처리시스템, 컴퓨터망 동일



하 성 룡

1981년 영남대학교 공과대학 전
자과(학사)
1983년 영남대학교 공과대학 전
자과 전자계산기전공(공학석
사)
1994년 영남대학교 공과대학
전산공학과 박사과정 수료

1984년~현재 경북전문대학 부교수

관심분야: 운영체제, 분산처리시스템, 컴퓨터망 동일



김 종 근

1981년 영남대학교 공과대학 전
자과(학사)
1987년 영남대학교 공과대학 전
자과 전자계산기전공(공학석
사)
1991년 (일본) 전기통신대학 정
보공학과 소프트웨어전공(공
학박사)

1984년~91년 경북전문대학 전산과 전임

1991년~현재 영남대학교 전산공학과 부교수

관심분야: 운영체제, 분산처리시스템, 컴퓨터망, 성능
평가 동일