집중형 분산처리시스템의 성능평가를 위한 퍼지 폐쇄형 BCMP 큐잉네트워크모델

추 봉 조[†]·조 정 복[††]·우 종 호[†††]

요 약

본 논문에서는 네트워크환경에 따라 모호성을 갖는 집중형 분산처리시스템의 성능평가를 위해 퍼지이론을 적용한 퍼지 폐쇄형 BCMP 큐잉네트워크모델을 제안하였다. 이 모델은 퍼지요소들을 처리할 수 있는 퍼지평균값분석방법을 사용하여 작업소요시간, 시스템처리율, 시스템내 작업수 및 서버활용율 등의 시스템성능을 평가할 수 있는 측도를 유도하였다. 이들의 유효함을 검증하기 위하여 퍼지서비스요구시간을 갖는 집중형 분산처리시스템에 클라이언트의 수의 변화에 따른 유도된 성능평가측도를 시뮬레이션하고, 그 결과를 고찰하였다. 제안된 모델은 모호성을 갖는 시스템의 성능을 평가할 때 기존의 방법보다 유연하고 실제적인 방법을 제공한다.

Fuzzy Closed BCMP Queueing Network Model for Performance Evaluation of Centralized Distributed Processing System

Bong Jo Choo[†]·Jung Bok Jo[††]·Chong Ho Woo[†††]

ABSTRACT

This paper proposes the fuzzy closed BCMP queueing network model using fuzzy set theory for the performance evaluation of centralized distributed processing system with ambiguous system factors in the network environments. This model can derive the measures for system performances such as the job spending time, the system throughput, average job number and server utilizations using fuzzy mean value analysis which can process the fuzzy factors. Computer simulation has been performed centralized distributed system with fuzzy service requirement time for verifying the effectiveness of derived equations of performance evaluation according to the numbers of clients, and the results were analyzed. The proposed model provides more and flexible realistic than performance evaluation of conventional method when we evaluated system performance with ambiguous factors.

키워드 : 큐잉네트워크모델(Queueing network model), 퍼지 BCMP 큐잉네트워크(Fuzzy BCMP queueing network), 분산처리시스템(Distributed processing system), 성능평가(System performance evaluation)

1. 서 론

분산처리시스템 및 네트워크시스템에서 성능의 평가는 작업량의 증감에 따른 시스템의 문제점 파악과 시스템 업그레이드 시기를 결정할 때 중요한 정보를 제공한다. 이러한 시스템 성능을 평가하는 방법에는 시스템을 수학적 해석모델로 변환하여 각종 성능평가측도들을 정량적으로 계산하는 큐잉네트워크모델이 있다[1, 2]. 이것은 작업의 도착율, 서버의 서비스율 및 작업의 서비스 요구 등에 임의의 확률분포를 가정하고, 시스템 외부로부터 작업진입여부에 따라 개방형, 폐쇄형 및 혼합형으로 구분하여 시스템의 성능을 평가한다. 특히, 폐쇄형 BCMP(Baskett, Chandy, Muntz and Palacios) 큐잉네트워크모델[2]은 시스템 외부로부터 작업진입이 없고, 작업들은 서비스형태에 따라 서로 다른 서비스요구와 시간분포를 가지고 분산처리시스템에서 클라이언트의 수와 서버의 수, 그리고 사용자평균작업동작시간 등을 고려하여 시스템의 성능을 평가한다.

한편, 큐잉네트워크모델에서 작업의 서비스요구와 서버의 서비스율은 요구한 작업형태, 통신라인의 상태, 장치간의 부하, 네트워크 트래픽 등에 따라 모호성(ambiguity)을 보이거나 새로운 시스템을 설계할 때에는 기존의 시스템에서 수집

† 정 회 원 : 김천대학 컴퓨터정보처리계열 교수
†† 정 회 원 : 동서대학교 인터넷공학부 컴퓨터및인터넷공학전공
††† 정 회 원 : 부경대학교 전자컴퓨터정보통신공학부 교수
논문접수 : 2001년 8월 1일, 심사완료 : 2001년 11월 7일

된 정보가 유용하지 못할 수도 있다. 이러한 경우에는 기존의 큐잉네트워크모델에서 각 요소에 임의의 확률분포를 적용하여 시스템의 성능을 평가하는 것보다는 언어적(linguistic) 표현의 퍼지집합이론을 적용하여 시스템의 성능을 분석을 하는 것이 보다 적합하여 다양한 퍼지 큐잉네트워크모델이 제안되었다[3-5]. 따라서 분산처리시스템에서 시스템외부로부터 작업의 진입이 없고, 작업이 서로 다른 서비스요구를 갖는 경우의 시스템 성능평가는 기존의 폐쇄형 BCMP 큐잉네트워크모델로 가능하나, 네트워크환경에 따라 모호성을 갖는 경우에는 시스템의 성능을 정확히 평가할 수 없다.

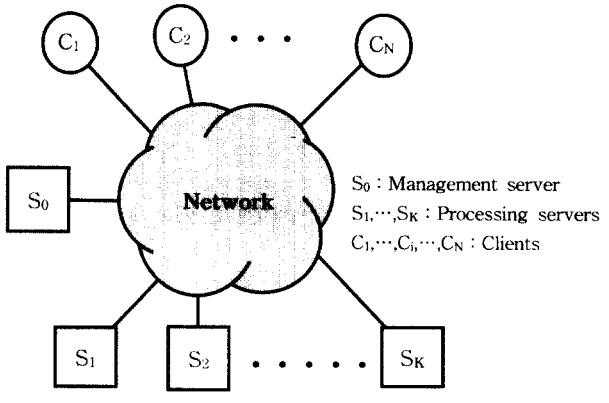
본 논문에서는 작업들이 서로 다른 서비스요구를 갖는 분산처리시스템에서 네트워크환경에 따라 시스템의 각 요소들이 모호성을 가질 경우, 퍼지이론을 적용하여 유연하고 실제적인 성능평가가 가능한 퍼지 폐쇄형 BCMP 큐잉네트워크모델을 제안한다. 제안된 모델의 성능평가는 기존의 평균값 분석방법[6, 7]에 퍼지요소들을 처리할 수 있는 퍼지평균값분석방법을 적용하여 작업소요시간, 시스템처리율, 시스템내 작업수 및 서버활용율 등을 구하였다. 유도된 성능평가측도의 검증은 4개의 서버와 작업이 서로 다른 서비스요구시간을 갖는 집중형 분산처리시스템에 수치적인 실제 값을 적용하여 클라이언트의 수에 따른 성능을 시뮬레이션하고, 그 결과를 고찰하였다. 제안된 모델은 도착한 작업이 서로 다른 서비스시간분포와 모호성을 갖는 분산처리 및 네트워크시스템의 성능을 평가 할 때 기존의 방법보다 유연하고 실제적인 방법이 될 수 있다.

2. 집중형 분산처리시스템과 BCMP 큐잉네트워크모델

2.1 집중형 분산처리시스템

분산처리시스템은 컴퓨터기술과 네트워크기술의 발달로 인해 네트워크를 통해 각종 서버, 주변장치, 허브, 스위치, 라우터 등 많은 장치들이 서로 유기적으로 연결되어 요구된 작업을 동시에 처리하는 구조이다. 이것은 작업처리관점에 따라 호스트기반처리, 주종속처리, 클라이언트/서버처리, 그리고 동배간처리 등으로 나눌 수 있다[8, 9]. 또한 노드의 부하를 균등하게 재분배시키는 부하균등방법에 따라 집중형과 분산형으로 나눌 수 있다[10, 11]. 분산형은 각 노드마다 부하테이블을 유지하여 신뢰도가 높으나 부하테이블의 일치성을 유지하기 위해 높은 통신오버헤드를 요구한다. 집중형은 임의의 관리서버를 두어 네트워크에 연결된 클라이언트에서 요구된 작업을 서버에 균등하게 재분배한다.

일반적인 클라이언트/서버모델의 집중형 분산처리시스템 구성은 (그림 1)과 같다. 여기서 관리서버 S_0는 네트워크를 통해 서버의 부하를 균등하게 분배하고, K개의 서버($S_1,\cdots,S_K$)는 각종 응용처리서버, 파일서버, 데이터베이스서버, 프린터서버 등이 되며, N개의 클라이언트($C_1,\cdots, C_n$)는 서로 다른 서비스요구와 서비스시간분포를 갖는다. 따라서 작업은 네트워크에 연결된 N개의 클라이언트에서 적절한 네트워크경로를 통해 분산처리시스템에게 서비스를 요구하고, 요구된 작업은 관리서버를 통해 이를 처리할 수 있는 K개의 서버에 분배된다. K개의 각 서버에 도착한 작업들은 해당 서버의 서비스를 받고 처리결과는 요청한 클라이언트에게 전송된다.



(그림 1) 클라이언트/서버 집중형 분산처리시스템의 구성

2.2 BCMP 큐잉네트워크모델

큐잉네트워크모델은 컴퓨터시스템, 네트워크시스템, 분산처리시스템 및 멀티컴퓨팅시스템 등의 성능을 평가하기 위해 사용되는 수학적 해석모델이다[1-5]. BCMP 큐잉네트워크모델[2]은 잭슨(Jackson) 큐잉네트워크[12]를 일반화한 것으로 작업이 서로 다른 서비스요구와 서비스시간분포를 갖는 시스템의 성능평가에 사용된다. 따라서 외부로부터 작업의 진입이 없고 작업이 서로 다른 서비스요구와 시간분포를 갖는 분산처리시스템의 성능평가는 폐쇄형 BCMP 큐잉네트워크모델로 가능하다.

폐쇄형 BCMP 큐잉네트워크모델에서 각 서버들은 FCFS 및 처리기공유 형태를 갖고, 클라이언트는 서로 다른 서비스요구와 서비스시간분포를 갖는다. 여기서 서버에 도착하는 작업의 도착율은 λ이고, 관리서버와 K개의 서버들의 서비스율은 각각 μ_0와 $\mu_1,\cdots,\mu_K$이다. 또한 관리서버의 서비스를 받은 작업은 p_0의 확률로 클라이언트에 출력되거나 각각 $p_1, p_2,\cdots,$ p_K의 분기확률로 K개의 서버에 진입하여 서비스를 받는다.

폐쇄형 BCMP 큐잉네트워크모델의 성능평가는 Raiser 등[6, 7]에 의해 증명된 수치적인 안정성과 해석의 용이성을 제공하는 평균값분석을 사용한다. 이 방법은 도착정리(arrival theorem)와 리틀의 법칙(Little's law)[13]을 근거로 작업평균소요시간, 시스템처리율, 시스템 내에 있는 작업수에 대한 평균값을 구하여 시스템의 성능을 평가한다. 특히, 시스템처리율은 시스템외부로부터 작업의 진입이 없기 때문에 클라이언트에서 입력된 작업이 서버로부터 서비스를 받고 출력되어 다시 서버에 입력되는 작업도착율이다[1, 2, 6, 7].

3. 퍼지집합이론과 퍼지리틀의 법칙

3.1 퍼지집합이론

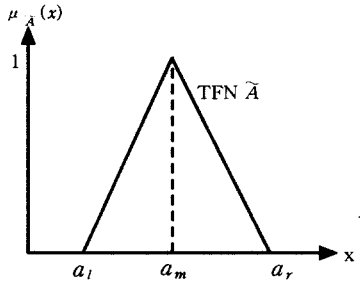
큐잉네트워크모델에서 작업의 서비스요구, 서버의 서비스율 등이 네트워크 환경에 따라 모호성을 가질 경우, 기존의 확률이론이나 크리스프(crisp) 집합으로 이들을 표현하는 것보다는 언어적인 표현의 퍼지집합이론을 적용하는 것이 보다 적합하다[3-5]. 본 논문에서는 모호성을 퍼지집합으로 나타내는 방법 중 계산의 효율성과 단순성을 갖는 삼각형퍼지수(triangular fuzzy number : TFN)를 사용한다[14-16].

퍼지집합 $\widetilde{A}$는 임의의 요소 x에 의해 일반화되는 개체의 집합 X에 대한 순서쌍(ordered pair)들의 집합으로, 식 (1)과 같다.

$$\widetilde{A} = \{(x, \mu_{\widetilde{A}}(x)) \mid x \in X,\ \mu_{\widetilde{A}}(x) \in [0,1]\} \tag{1}$$

여기서 $\mu_{\widetilde{A}}(x)$는 x에 대한 퍼지소속함수이다.

임의의 요소 x에 대한 퍼지집합 $\widetilde{A}$를 TFN으로 나타내면 (그림 2)와 같이 3개의 값 $\langle a_l, a_m, a_r \rangle$ 즉, 왼쪽(left), 중간(middle), 오른쪽(right)의 값들로 표현되며, 이들의 소속함수는 식 (2)와 같다.



(그림 2) TFN $\widetilde{A}$의 3개 $\langle a_l, a_m, a_r \rangle$값의 범위

$$\mu_{\widetilde{A}}(x) = \begin{cases} 0 & \text{for } x < a_l \\ \dfrac{x - a_l}{a_m - a_l} & \text{for } a_l \le x \le a_m \\ \dfrac{a_r - x}{a_r - a_m} & \text{for } a_m \le x \le a_r \\ 0 & \text{for } x > a_r \end{cases} \tag{2}$$

TFN $\widetilde{A} = \langle a_l, a_m, a_r \rangle$과 또 다른 TFN $\widetilde{B} = \langle b_l, b_m, b_r \rangle$에 대한 가산과 승산은 다음의 식 (3) 및 식 (4)와 같다.

$$\begin{aligned} \widetilde{A} + \widetilde{B} &= \langle a_l, a_m, a_r \rangle + \langle b_l, b_m, b_r \rangle \\ &= \langle a_l + b_l,\ a_m + b_m,\ a_r + b_r \rangle \end{aligned} \tag{3}$$

$$\begin{aligned} \widetilde{A} \cdot \widetilde{B} &= \langle a_l, a_m, a_r \rangle \cdot \langle b_l, b_m, b_r \rangle \\ &= \langle a_l b_l, a_m b_m, a_r b_r \rangle \end{aligned} \tag{4}$$

그리고 위의 두 개의 TFN $\widetilde{A}$, $\widetilde{B}$에 대한 퍼지방정식 $\widetilde{X} \div \widetilde{A} = \widetilde{B}$의 해 $\widetilde{X}$는 역함수를 이용하여 식 (5)와 같이 구할 수 있다.

$$\widetilde{X} = \widetilde{B} \odot \widetilde{A} = \langle b_r a_l, b_m a_m, b_l a_r \rangle \tag{5}$$

$$iff\ b_l a_r < b_m a_m < b_r a_l\ when\ \widetilde{A} > 0,\ \widetilde{B} \ge 0$$

3.2 퍼지 리틀의 법칙

리틀의 법칙은 큐잉이론을 이용하여 시스템의 성능분석을 할 경우에 사용되는 가장 기본적인 법칙으로 시스템으로 들어오는 작업의 도착율과 시스템 내에 있는 장치의 서비스율을 알면 현재 시스템 내에 있는 작업수와 작업의 평균소요시간을 계산할 수 있다[13]. 한편, Jo와 Yamazaki[3]는 기존의 리틀의 법칙에 비결정적이고 모호한 작업의 도착율과 장치의 서비스율에 퍼지집합이론을 적용한 퍼지리틀의 법칙을 제안하였다.

시스템 내에서 t 시각에 존재하는 작업수, $\widetilde{N}(t)$는 시스템으로 도착한 작업수 $\widetilde{A}(t)$와 서비스를 받고 시스템을 떠난 작업수 $\widetilde{D}(t)$를 이용하여 나타내면 식 (6)과 같다.

$$\widetilde{N}(t) = \widetilde{A}(t) - \widetilde{D}(t) \tag{6}$$

따라서 $[0, t]$ 시간까지 시스템 내에 존재하는 작업수, $\widetilde{N}_t$는 식 (7)과 같다.

$$\widetilde{N}_t = \frac{1}{t} \int_0^t \widetilde{N}(t)dt \tag{7}$$

이 식은 시간 t가 증가함에 따라 시스템은 시간의 흐름에 독립적인 안정상태가 되고, 이때의 시스템 내에 존재하는 작업수 $\widetilde{N}$은 식 (8)과 같이 정의할 수 있다.

$$\widetilde{N} = \lim_{t \to \infty} \widetilde{N}_t \tag{8}$$

또한, $[0, t]$ 시간 사이에 시스템으로 도착한 작업의 퍼지도착율 $\widetilde{\lambda}_t$는 식 (9)와 같고 안정상태일 때, 작업의 퍼지도착율 $\widetilde{\lambda}$는 식 (10)과 같다.

$$\widetilde{\lambda}_t = \frac{\widetilde{A}(t)}{t} \tag{9}$$

$$\widetilde{\lambda} = \lim_{t \to \infty} \widetilde{\lambda}_t \tag{10}$$

따라서 $[0, t]$ 시간까지 작업이 시스템에 도착하여 서비스를 받고 떠나기까지 소요되는 평균소요시간 $\widetilde{W}_t$와 안정상태의 $\widetilde{W}$는 각각 식 (11)과 식 (12)가 된다.

$$\widetilde{W}_t = \frac{1}{\widetilde{A}(t)} \int_0^t \widetilde{N}(t)\,dt \tag{11}$$

$$\widetilde{W} = \lim_{t \to \infty} \widetilde{W}_t \tag{12}$$

그러므로 최종적으로 시스템이 안정상태일 때, 작업의 평균 소요시간 $\widetilde{W}$는 작업의 퍼지도착율 $\widetilde{\lambda}$와 시스템내 존재하는 작업수 $\widetilde{N}$로 나타내면 식 (13)과 같다.

$$\widetilde{W} = \frac{\widetilde{N}}{\widetilde{\lambda}} \tag{13}$$

4. 퍼지 폐쇄형 BCMP 큐잉네트워크모델을 이용한 시스템 성능평가

네트워크 및 시스템환경에 따라 모호성을 갖는 집중형 분산처리시스템의 성능평가를 위해 퍼지집합이론을 적용한 퍼지 폐쇄형 BCMP 큐잉네트워크모델을 제안하고, 퍼지 파라메터를 처리할 수 있는 퍼지평균값분석을 통하여 작업평균소요시간, 시스템처리율, 시스템 내에 작업수 및 서버활용율 등 성능평가측도를 유도한다.

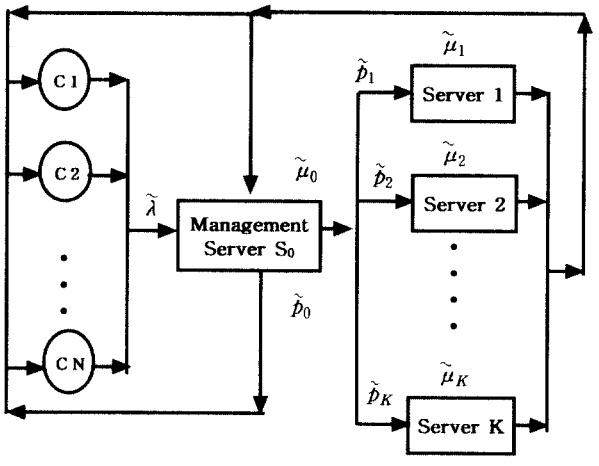
4.1 집중형 분산처리시스템의 퍼지 폐쇄형 BCMP 큐잉네트워크 모델화

집중형 분산처리시스템에서 서버의 서비스율과 작업의 서비스요구 등은 어떤 임의의 확률분포로 표현될 수도 있으나, 통신라인의 상태, 서버의 상태, 서버에 요구한 작업형태, 네트워크에 연결된 장치간의 부하 등 네트워크환경에 따라 비결정적인 다양한 언어적인 형태로 표현될 수도 있다. 이러한 경우에는 확률분포를 사용하여 시스템의 성능을 평가하는 것 보다 모호함을 표현하고 처리할 수 있는 퍼지집합이론을 적용하여 시스템의 성능을 평가하는 것이 보다 적합하다.

시스템성능분석에 퍼지이론을 적용함으로 각 요소에 임의의 확률적 분포를 강요할 필요가 없고, 확률적인 선택에서 발생하는 오류를 줄일 수 있다. 또한 시스템 성능평가에 단일적이고 결정적인 값을 제공하기보다는 퍼지 소속함수를 통해 성능의 정도에 따른 구간 값을 제공할 수 있어 시스템의 성능을 보다 유연하고 실제적으로 평가할 수 있다.

$K+1$개의 서버와 N개의 클라이언트로 구성된 집중형 분산처리시스템은 (그림 3)의 퍼지 폐쇄형 BCMP 큐잉네트워크모델로 표현될 수 있다.

여기서 작업들은 네트워크에 연결된 N개의 클라이언트에서 각 서버에 다른 서비스요구로 요청되어 하나의 관리서버 S_0에 도착된다. 관리서버에 도착한 작업들은 적절한 네트워크경로를 따라 다른 서비스율을 갖는 K개의 각 서버에 할당되어 서버의 서비스를 받고, 그 결과는 관리서버를 통해 네트워크나 클라이언트로 전송된다. 여기서 표시된 패스들은 고정된 경로를 갖는 것이 아니라 네트워크의 상황에 따라 통신시스템에 의해서 설정된다. 또한 각 퍼지 파라메터는 네트워크 및 시스템 환경에 따라 '높다', '보통이다', '낮다' 등의 모호한 언어적 표현을 갖는 TFN이며 다음과 같은 의미이다.



(그림 3) 퍼지 폐쇄형 BCMP 큐잉네트워크모델

(1) $\widetilde{\lambda}$: 네트워크에 연결된 클라이언트에서 시스템으로 도착하는 작업의 퍼지도착율

(2) $\widetilde{\mu}_0$: 네트워크에 연결된 서버의 관리를 담당하는 관리서버의 퍼지서비스율

(3) $\widetilde{\mu}_i$: 도착한 작업을 처리하는 서버 i의 퍼지서비스율 $(1 \le i \le K)$

(4) $\widetilde{p}_0$: 관리서버의 서비스를 받고 클라이언트로 출력하는 퍼지확률

(5) $\widetilde{p}_i$: 도착한 작업이 서버 i로 진입하는 퍼지분기확률 $(1 \le i \le K)$

4.2 퍼지 폐쇄형 BCMP 큐잉네트워크모델의 성능평가

퍼지 폐쇄형 BCMP 큐잉네트워크모델에서 성능평가측도는 기존의 평균값분석을 근거로 퍼지 파라메터를 처리할 수 있는 퍼지평균값분석으로 유도한다. 퍼지평균값분석은 작업이 각 서버에 서로 다른 퍼지서비스요구시간 $\widetilde{D}_k$를 가질 때, 다음의 작업평균소요시간, 시스템처리율, 시스템 내에 존재하는 작업수 및 서버활용율 등을 유도한다.

4.2.1 작업의 평균소요시간 $\widetilde{W}$

작업의 평균소요시간은 각 클라이언트에서 요청한 작업이 서버에 도착되어 작업의 처리가 완료될 때까지의 시스템

에서 소요하는 시간이다. 따라서 클라이언트의 수가 n일 때, 서버 k에 도착한 작업이 서비스를 받고 떠나기까지 소요되는 평균소요시간, $\tilde{W}_k[n]$은 도착정리에 의해 식 (14)와 같다.

$$\tilde{W}_k[n] = \tilde{D}_k + \tilde{D}_k \cdot \tilde{L}_k[n-1] \quad \text{for } k = 0,1,\cdots,K, \text{ for } n = 1,2,\cdots,N \tag{14}$$

여기서 $\tilde{D}_k$는 작업이 서버 k에게 요구한 서비스요구시간이며, $\tilde{L}_k[n-1]$는 지금 n번째 클라이언트에서 도착한 작업 이전에 서버 k에 존재하는 작업수를 나타낸다. 그러므로 서버 k에 도착한 작업의 평균소요시간은 지금 도착한 작업 이전에 서버 k에 존재하는 작업들이 모두 서비스를 받을 때까지의 시간과 자신이 요구한 서비스요구시간의 합이 된다. 따라서 클라이언트의 수가 n일 때, 모든 서버에 대한 작업의 평균소요시간은 식 (14)에서 구한 각 서버들의 작업소요시간의 합인 식 (15)와 같다.

$$\tilde{W}[n] = \sum_{k=0}^{K} \tilde{W}_k[n] \tag{15}$$

그러므로 클라이언트 수가 최종적으로 N개일 때, 시스템 전체에서 작업의 평균소요시간은 식 (16)과 같다.

$$\tilde{W} = \tilde{W}[N] \tag{16}$$

4.2.2 시스템 처리율 $\tilde{\lambda}$

따라서 클라이언트의 수가 $n(n=1,2,\cdots,N)$일 때, 시스템처리율은 클라이언트의 수 n에 평균소요시간 $\tilde{W}[n]$를 나누면 된다. 한편 폐쇄형 큐잉네트워크에서 사용자가 각 클라이언트를 통해 작업명령을 입력하기 위해 소요되는 평균시간, 즉 평균작업동작시간은 시스템처리율에 영향을 줄 수 있다. 따라서 사용자의 퍼지평균작업동작시간을 $\tilde{Z}$라고 하면, 클라이언트의 수가 n일 때, 시스템처리율 $\tilde{\lambda}[n]$은 식 (17)과 같다.

$$\tilde{\lambda}[n] = \frac{n}{\tilde{W}[n] + \tilde{Z}} \quad \text{for } n=1,2,\cdots,N \tag{17}$$

여기서 서버의 수가 고정이 되면 $\tilde{\lambda}[n]$은 클라이언트의 수에 의존됨으로 시스템에 요구된 처리율에 적합한 클라이언트의 수를 결정할 수가 있다. 평균작업동작시간은 기존의 폐쇄형 BCMP 큐잉네트워크모델에서는 사용자들이 클라이언트를 동작하는 시간의 평균값으로 가정을 하였지만, 퍼지 폐쇄형 BCMP 큐잉네트워크모델에서는 사용자의 동작특성상 평균작업동작시간을 '느리다', '보통이다', '빠르다' 등의 퍼지형태로 표현하여 계산한다. 그러므로 클라이언트의 수가 최종적으로 N개일 때, 시스템처리율은 식 (18)과 같다.

$$\tilde{\lambda} = \tilde{\lambda}[N] \tag{18}$$

4.2.3 시스템 내에 존재하는 작업수 $\tilde{L}$

시스템 내에 존재하는 작업수는 작업의 퍼지도착율과 작업의 평균소요시간을 알면 퍼지리틀의 법칙의 식 (13)을 이용하여 구할 수 있다. 그러나 퍼지집합이론에서는 나눗셈에 대한 직접적인 역함수가 존재하지 않기 때문에, $\odot$ 퍼지연산자를 이용하여 구한다. 따라서 클라이언트의 수가 n일 때, 서버 k내에 존재하는 작업수 $\tilde{L}_k[n]$는 식 (15)와 식 (17) 이용하여 식 (19)와 같이 구할 수 있다.

$$\tilde{L}_k[n] = \tilde{\lambda}[n] \odot \tilde{W}_k[n] \quad \text{for } k = 0,1,\cdots,K, \text{ for } n = 1,2,\cdots,N \tag{19}$$

그러므로 클라이언트의 수가 n일 때, 모든 서버에 존재하는 작업수, $\tilde{L}[n]$은 식 (19)에서 구한 값들의 합이 되며, 이는 식 (20)과 같다.

$$\tilde{L}[n] = \sum_{k=0}^{K} \tilde{L}_k[n] \tag{20}$$

따라서 클라이언트의 수가 최종적으로 N개일 때, 시스템 전체에서 작업수, $\tilde{L}$은 식 (21)과 같다.

$$\tilde{L} = \tilde{L}[N] \tag{21}$$

4.2.4 서버 활용율 $\tilde{\rho}_k$

서버 k의 서버활용율, $\tilde{\rho}_k$는 시스템에 도착한 작업에 대해 각 서버들의 활용율을 평가하는 측도로서 시스템처리율 $\tilde{\lambda}$에 대한 서버 k의 활용율을 나타낸다. 따라서 $\tilde{\rho}_k$는 식 (18)에서 구한 클라이언트의 수가 N일 때의 시스템처리율 $\tilde{\lambda}$와 서버 k에 요구한 작업의 퍼지서비스요구시간 $\tilde{D}_k$를 이용하여 식 (22)와 같이 구할 수 있다.

$$\tilde{\rho}_k = \tilde{\lambda} \cdot \tilde{D}_k \quad \text{for } k=0,1,\cdots,K \tag{22}$$

여기서 각 서버들의 활용율은 직관적으로 시스템처리율과 서버에 요구한 작업의 퍼지서비스요구시간이 높을수록 서버의 활용율은 높다는 것을 알 수 있다.

퍼지평균값분석으로 유도된 식 (14)에서 식 (22)까지의 성능평가식들을 이용한 퍼지 폐쇄형 BCMP 큐잉네트워크모델의 성능평가측도의 유도과정을 정리하면 (그림 4)와 같다.

Algorithm* : *Performance Measures of Fuzzy Closed BCMP Queueing Network Model

- *System has K+1 servers and N clients*
- $\tilde{\lambda}$: *system throughput for the system*
- $\tilde{D}_k$: *fuzzy service requirement time for server k*
- $\tilde{W}_k[n]$: *average job spending time for at server k*
- $\tilde{W}$: *average job spending time for the system*
- $\tilde{Z}$: *fuzzy mean job operation time for the user*
- $\tilde{L}_k[n]$: *number of jobs for n clients at server k*
- $\tilde{L}$: *number of jobs for the system*
- $\tilde{\rho}_k$: *utilization of the server k*

Step 1. Initialize

$$Set\ \tilde{L}_k[0] = 0, \quad k = 0,1,2,\cdots,K$$

$$Set\ \tilde{W} = \tilde{L} = 0, \quad n = 1,2,\cdots,N$$

Step 2. Iterate for n=1,2,…,N

$$\tilde{W}_k[n] = \tilde{D}_k(1 + \tilde{L}_k[n-1]), \quad k = 0,1,\cdots,K$$

$$\tilde{W}[n] = \sum_{k=0}^{K} \tilde{W}_k[n]$$

$$\tilde{\lambda}[n] = \frac{n}{\tilde{W}[n] + \tilde{Z}}$$

$$\tilde{L}_k[n] = \tilde{\lambda}[n] \odot \tilde{W}_k[n], \quad k = 0,1,\cdots,K$$

$$\tilde{L}[n] = \sum_{k=0}^{K} \tilde{L}_k[n]$$

Step 3. Set system performance measures with K+1 Servers and N clients

$$Set\ \tilde{\lambda} = \tilde{\lambda}[N]$$

$$Set\ \tilde{W} = \tilde{W}[N]$$

$$Set\ \tilde{L} = \tilde{L}[N]$$

Step 4. Compute Utilizations of Servers

$$\tilde{\rho}_k = \tilde{\lambda} \cdot \tilde{D}_k, \quad k = 0,1,\cdots,K$$

(그림 4) 퍼지 폐쇄형 BCMP 큐잉네트워크의 성능평가측도의 유도과정

5. 시뮬레이션 및 결과고찰

제안된 퍼지 폐쇄형 BCMP 큐잉네트워크모델에서 유도한 성능평가식들의 유효함을 확인하기 위하여 수치적인 예를 적용하고, 그 결과를 분석한다. 여기서 시스템에 도착한 작업이 서버 k에 요구하는 퍼지서비스요구시간 $\tilde{D}_k$가 '높다', '보통이다', '낮다' 등의 언어적인 표현에 대한 TFN 값을 사용할 때, 유도된 시스템성능평가측도를 계산하였다.

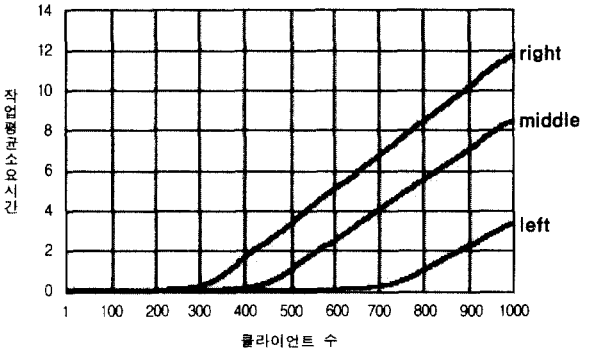
[수치 예] 집중형 분산처리시스템 성능분석(4개 서버와 1~1000개 클라이언트로 구성)

집중형 분산처리시스템은 총 4개의 서버를 가지며, 작업이 각 서버에 요구하는 퍼지서비스요구시간 $\tilde{D}_k$는 <표 1>과 같다. 이것은 각 서버에 대해 서비스요구시간이 '높다', '보통이다', '낮다'의 요구시간을 TFN 값으로 표현하였다. 또한 사용자가 클라이언트를 동작시키고 퍼지평균작업동작시간 $\tilde{Z}$는 '느리다', '보통이다', '빠르다' 등으로 표현될 수 있으며, $\tilde{Z}$의 TFN 값을 $\tilde{Z}$=<5.2, 6.5, 8.6>으로 가정하여 클라이언트 수의 변화에 따른 작업의 평균소요시간, 시스템처리율, 시스템내 작업수 및 서버활용을 계산하였다.

<표 1> 각 서버의 퍼지서비스요구시간

서 버	퍼지서비스요구 $\tilde{D}_k$
0	<0.012, 0.015, 0.017>
1	<0.010, 0.012, 0.014>
2	<0.004, 0.006, 0.009>
3	<0.003, 0.005, 0.006>

(그림 5)는 작업의 퍼지서비스요구시간이 <표 1>과 같을 때, 클라이언트의 수가 1에서 1000까지 변화할 때의 작업의 평균소요시간의 TFN 변화를 나타낸다. 여기서 평균소요시간 TFN의 오른쪽 값의 변화를 살펴보면, 평균소요시간은 클라이언트의 수가 1에서 250정도까지는 서서히 증가하다가 이후부터는 급격한 증가를 보였다. 이는 클라이언트의 수가 증가할 수록 도착정리에 의해 시스템 내에 작업수가 점점 많아지게 되고, 어느 순간이 되면 시스템처리율에 비해 시스템 내에 존재하는 작업수가 너무 많아져 평균소요시간도 급격한 증가를 보인다.

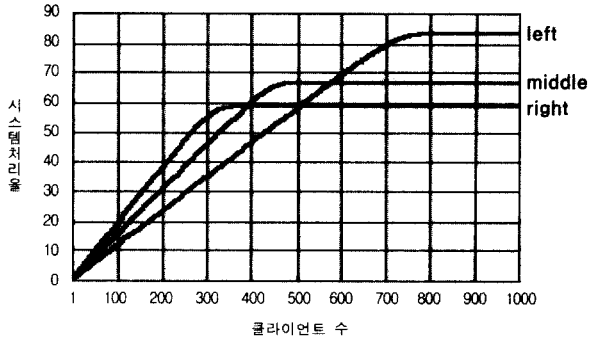


(그림 5) 클라이언트 수의 변화에 따른 작업평균소요시간(TFN)

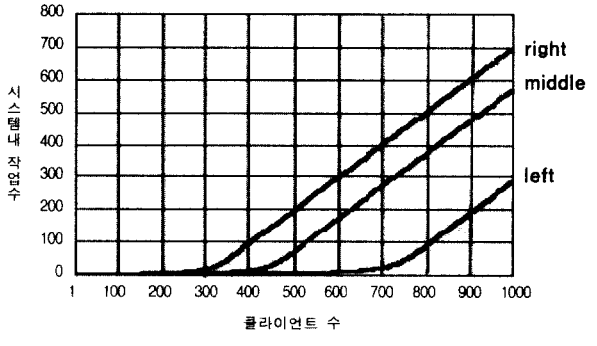
(그림 6)은 식 (17), 식 (18)을 통해 유도한 시스템처리율에 대한 TFN의 변화이다. 여기서 시스템처리율은 클라이언트의 수가 증가할수록 일정하게 증가하다가, 어느 순간 이후부터는 큰 변화가 없이 일정한 값에 수렴됨을 알 수 있다. 이는 클라이언트의 수가 증가할 수록 시스템에 도착하는 작업수가 증가하게 되고, 어느 순간부터는 작업의 도착율이 시스템처리율보다 높아 더 많은 작업이 시스템에 들어오기 때문이다. 또한, 시스템처리율의 TFN에서 오른쪽 값과 왼쪽 값이 서로 교차되어 수렴함을 보이는 것은 (그림 5)의 작업평균소요시간 TFN에서 오른쪽 값이 왼쪽 값보다 더 적은 클라이언트 수에서 급격하게 증가하기 때문에 시스템처리율에서는 오른쪽 값이 먼저 수렴하였다.

(그림 7)은 시스템 내에 존재하는 작업수에 대한 TFN의

변화이다. 시스템 내에 존재하는 작업수는 작업평균소요시간과 마찬가지로 일정하게 증가하다가 갑자기 급격한 증가를 보였다. 이는 시스템 내에 존재하는 작업수는 평균소요시간과 시스템처리율에 의존하기 때문에 평균소요시간이 급격히 증가할 수록 작업수도 급격히 증가한다. 따라서 시스템 내에 존재하는 작업수는 평균소요시간이 클수록, 그리고 시스템처리율이 높을 수록 빠르게 증가한다.



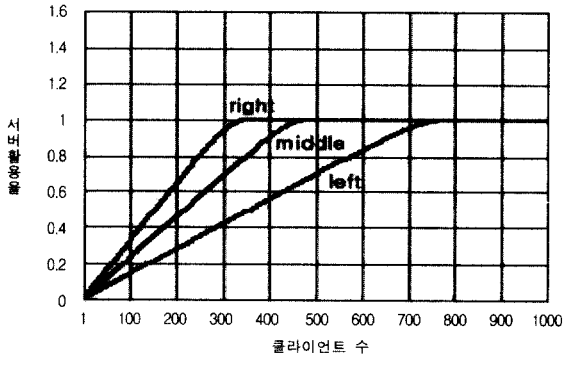
(그림 6) 클라이언트 수의 변화에 따른 시스템처리율(TFN)



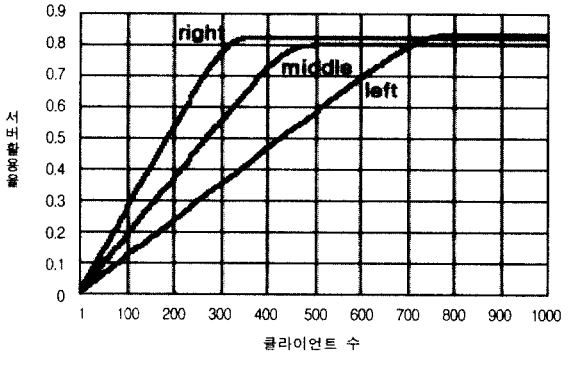
(그림 7) 클라이언트 수의 변화에 따른 시스템내 작업수(TFN)

(그림 8)과 (그림 9)는 서버 0과 서버 1에 대한 서버활용율의 TFN 변화를 나타낸 것이다. 서버 k의 활용율은 시스템처리율과 서버 k에게 요구한 서비스시간분포의 곱으로 유도되었다. 그러므로 서버처리율도 시스템처리율과 마찬가지로 클라이언트의 수가 증가할 수록 일정하게 증가하다가 어느 순간부터는 큰 변화가 없이 일정한 값에 수렴됨을 알 수 있다. 이는 클라이언트의 수가 증가할 수록 서버에 도착하는 작업수는 증가하게되고, 어느 순간부터는 작업도착율이 서버처리율보다 높아 더 많은 작업이 서버에 존재하기 때문이다. 그리고 나머지 서버들도 동일한 특성을 보였다.

시뮬레이션 결과, 한정된 서버의 수를 갖는 집중형 분산처리시스템에서 평균소요시간과 시스템 내에 존재하는 작업수는 클라이언트의 수가 증가함에 따라 서서히 증가하다가 어느 순간부터 갑자기 증가함을 알 수가 있었다. 또한 시스템처리율과 각 서버의 활용율은 클라이언트의 수가 증가함에 따라 일정하게 증가하다가, 어느 순간부터는 큰 변화가 없이 일정한 값에 수렴됨을 알 수 있다. 이는 기존의 폐쇄형 BCMP 큐잉네트워크와 동일한 특성을 보였다.



(그림 8) 클라이언트 수의 변화에 따른 서버 0의 서버활용율(TFN)



(그림 9) 클라이언트 수의 변화에 따른 서버 1의 서버활용율(TFN)

6. 결 론

네트워크환경에 따라 모호성을 갖는 집중형 분산처리시스템의 성능평가를 위해 기존의 폐쇄형 BCMP 큐잉네트워크모델에 퍼지이론을 적용하여 유연하고 실제적인 성능평가가 가능한 퍼지 폐쇄형 BCMP 큐잉네트워크모델을 제안하였다. 이 모델에서 퍼지요소들은 계산이 용이한 TFN으로 표현하고, 성능평가측도는 퍼지평균값분석을 적용하여 클라이언트 수의 변화에 대한 작업의 평균소요시간, 시스템의 처리율, 시스템내 작업수 및 서버의 활용율을 유도하였다. 유도한 성능평가측도의 유효함을 검증하기 위하여 작업의 퍼지서비스요구시간에 클라이언트의 수의 변화에 따른 성능평가의 주요식에 수치적인 실제 값을 적용하여 분석하였고, 그 결과는 다음과 같다.

(1) 작업의 평균소요시간 $\widetilde{w}$: 작업이 시스템내 도착하여 서비스가 완료될 때까지의 시간으로 클라이언트의 수가 증가함에 따라 서서히 증가하다가 어느 순간이후 급격히 증가한다.

(2) 시스템처리율 $\tilde{\lambda}$: 클라이언트의 수가 증가할수록 시스템처리율도 증가하다가, 어느 순간부터는 큰 변화가 없이 일정한 값에 수렴한다.

(3) 시스템내 작업수 $\tilde{L}$: 작업의 평균소요시간 $\tilde{W}$와 마찬가지로 클라이언트의 수가 증가함에 따라 서서히 증가하다가 어느 순간이후 급격한 증가한다.

(4) 서버활용율 $\tilde{\rho}_k$: 시스템처리율과 작업의 퍼지서비스요구시간의 곱으로 구하며, 클라이언트의 수가 증가할수록 서버의 활용율도 증가하다가, 어느 순간 이후부터 일정한 값에 수렴한다.

본 연구결과를 이용하면 퍼지 작업환경을 가진 제한된 자원과 서로 다른 서비스요구시간을 갖는 분산처리 및 네트워크시스템을 보다 실제적이고 유연하게 성능을 평가할 수 있다. 또한 시스템설계자가 시스템의 규모와 업그레이드 시기를 결정할 때, 원하는 시스템처리율과 작업소요시간을 갖는 적절한 클라이언트의 수와 서버의 수를 결정할 수 있다.

참 고 문 헌

[1] A. O. Allen, "Probability, Statistics, and Queueing Theory with Computer Science Applications," Academic Press, California, 1990.

[2] F. Baskett, K. M. Chandy, R. R. Muttz, and F. G. Palacios, "Open, closed, and mixed networks of queues with different classes of customers," J. ACM, Vol.22, pp.248-260, 1975.

[3] J. B. Jo, Y. Tsujimura, M Gen and G. Yamazaki, "Performance Evaluation of Computer System with Failure Based on Fuzzy Set Theory," Journal of the Operations Research Society of Japan, Vol.38, No.4, pp.409-422, 1995.

[4] 추봉조, 조정복, 우종호, "네트워크시스템의 성능평가를 위한 퍼지 M/M/1/K 큐잉네트워크 모델", 대한전자공학회논문지, 제38권, CI편, 제4호, pp.169-177, 2001.

[5] D. S. Negi and E. S. Lee, "Analysis and simulation of fuzzy queues," Fuzzy Sets and Systems, Vol.46, pp.321-330, 1992.

[6] M. Reiser, and S. S. Lavenberg, "Stationary state probabilities at arrival instants for closed queueing network with multiple types of customers," Advanced Application Problems, Vol.17, pp.1048-1061, 1980.

[7] I. F. Akyidiz, "Mean value analysis approximation for multiple server queueing network," Performance Evaluation, Vol.9, No.2, pp.77-91, 1988.

[8] J. Davidson, "Parallel & Distributed Processing," SIGCSE Bulletin Computer Science Education, 1998.

[9] J. B. Stefani, "Open distributed processing : an architectural basis for information networks," Computer communications, Vol.18, No.11, pp.849-862, 1995.

[10] A. Barak and A. Shillon, "A distributed load balancing policy for a multi computer," Software Pract. Expert, pp.901-913, 1985.

[11] T. Holzer, "Performance Measurement of Task allocation in a Distributed System," Ph. D. Thesis, Victoria Univ. 1996.

[12] J. R. Jackson, "Networks of waiting lines," Operations Research, Vol.5, No.4, pp.518-521, 1957.

[13] John D. C. Little, "A proof of the queueing formula : $L = \lambda W$," Operations Research, Vol.9, No.3, pp.383-387, 1961.

[14] A. Kaufmann and M. M. Gupta, "Introduction to Fuzzy Arithmetic," Van Nostrand Reinhold, 1985.

[15] H. J. Zimmermann, *Fuzzy Set Theory and Its Applications*, 2nd ed., Kluwer Academic Publishers, 1991.

[16] J. J. Buckley and Y. Qu, "Solving Systems of Fuzzy Equations : A New Solution Concept," Fuzzy Sets and Systems, Vol.39, pp.291-301, 1991.

추 봉 조

e-mail : bjchoo@kimcheon.ac.kr

1990년 경성대학교 전산통계학과 졸업(이학사)

1992년 경성대학교 대학원 전자계산학과 졸업(이학석사)

1996년 부경대학교 대학원 전자공학과 박사과정 수료

1996년~현재 김천대학 컴퓨터정보처리계열 조교수

관심분야 : 병렬처리, 네트워크성능평가,

조 정 복

e-mail : jobok@kowon.dongseo.ac.kr

1978년 경북대학교 전자계산기공학과 졸업(공학사)

1986년 영남대학교 전자공학과 졸업(공학석사)

1995년 일본 동경도립과학기술대학 제어정보시스템전공 졸업(공학박사)

1980년~1992년 경남전문대학 전자계산과 조교수

1992년~현재 동서대학교 인터넷공학부 컴퓨터 및 인터넷공학전공 조교수

관심분야 : GA, 퍼지큐잉시스템

우 종 호

1978년 경북대학교 전자공학과(전자계산기공학모듈) 졸업(공학사)

1981년 경북대학교 대학원 전자공학과 전산공학전공졸업(공학석사)

1990년 경북대학교 대학원 전자공학과 전산공학전공졸업(공학박사)

1978년~1981년 경남전문대학 전자계산과 전임강사

1981년~1996년 부산수산대학교 전자공학과, 컴퓨터공학과 교수

1996년~현재 부경대학교 전자컴퓨터공학부 교수

1987년~1988년 미국 뉴욕, RPI CS, 객원교수

2001년~현재 미국 루이지애나, LSU Baton Rouge ECE, 객원교수

관심분야 : 병렬처리시스템, VLSI 어레이, 인터넷 응용