〈研究論文〉

顧客이 任意数의 Server를 원하는 M/M/s sytem의 概算法

(An approximation of the M/M/s system where customers demand random number of servers)

金 聖 植*

Abstract

In the case of numerical implementation, the exact solution method for the M/M/s system where customers demand multiple server use [2] reveals limitations, if a system has large number of servers or types of customers. This is due to the huge matrices involved in the course of the calculations. This paper offers an approximation scheme for such cases. Capitalizing the characteristics of the service rate curve of the system, this method approximates the service rate as a piecewise linear function. With the service rates obtained from the linear function for each number of customers n (n=0, 1, 2,…), $\mu(n)$, steady-state probabilities and measures of performance are found treating this system as an ordinary M/M/s system. This scheme performs well when the traffic intensity of a system is below about 0.8. Some numerical examples are presented.

서 론

M/M/s system 에서 고객들이 이들의 type에 따라 각각 다른수의 server를 요구하는 경우에 대한 연구는 [2]에서 다루어져 정확한 결과(exact solution)를 구하는 방법이 제시되었다. 이 방법에서는 state 간의 transition들이 행렬로 나타내어져 이 transition probability matrix (T. P. M.), T, 가 해결되어야 할 문제의 크기 (size)를 결정하게 된다. 이 system의 state들은 다차원 (Multi-dimension)으로 표시되고 따라서 고려되어야 할 state의 수가 많으므로 상기 방법은 실제 적용시 대부분 T. P. M. T의 submatrice**의 order가 큰 문제를 다루어야 하나 이는 계산시 어려움이 따르는 문제인 것이다. 이러한 문제는 본 system 또는 상기 방법에서만 야기되는 문제가 아니고 state들이 다수의 변수 (Multivariate)로 표시되어져야 할 대기행렬에서는 어떠한 방법을 적용하던 공통적으로 발생하는 문제이다.

본 system에서 이제 s를 server의 수, K를 고객의 type의 수, d_k ($d_1 < d_2 < \cdots < d_k \leqq s$)를 k번째 type의 고객이 원하는 server의 수라하자. 또 U를 s/d_1보다 작은 정수 (integer) 중 가장 큰 수라고 하면 T. P. M. T의 가장 큰 submatrix order는 K^U에 의하여 결정된다.*** 따라서 K 또는 U가 증가함에 따라 계산시 필연적으로 다루어져야 하는 행렬의 크기는 지수 (ex-

* 고려대학교

본 연구는 과학재단의 지원에 의해 이루어진 것임.

** 행렬 T의 형태는 이의 대각선 상에 Non-zero Submatrix들이 있고 나머지 부분은 모두 0인 행렬이다. 식 (2) 참조.

*** system의 성질에 대하여 자세한 사항은 [1]의 page 2를 참조.

ponentially) 배로 급격히 증가하게 된다. 계산과정을 computer로 수행할때 행렬 T의 submatrix들은 항상 기억되어 있어야 하므로 계산가능한 문제의 크기는 주어진 computer의 기억용량에 의하여 제한을 받게된다. 예로서 C. D.C Cyber 72의 경우 가장 큰 submatrix의 order가 대략 120이 되면 〔2〕의 방법은 계산이 불가능해진다. 이에 행렬 T의 크기에 상관없이 계산할 수 있는 방법의 필요성이 대두하게 되어 본 연구는 다소 결과의 정확도를 희생하더라도 항상 계산이 가능한 개산(approximation)방법을 제시하는데 목적을 두고 있다. 정확한 계산법으로는 system 내의 고객수 및 이들의 type을 동시에 표시할 수 있도록 state들이 정해지며 이러한 state들의 stationary state probability가 계산된다. 이때 행렬 T의 크기는 허용된 고객의 type의 수에 의하여 결정되고 동시에 type들은 T에 반영된다. 다시 말해 행렬 T를 이용하지 않고는 고객의 type들이 표시된 state들의 state probability를 계산할 수 없다. 여기서 소개될 개산법은 T의 크기 때문에 고안된 방법이므로 이의 크기에 영향을 받지 않기 위해서는 type들은 고려하지 않게되고 단지 시스템 내에 n 명이 있을 확률을 구하는데 그 목표를 두고 있다. 이 연구는 〔1〕에서 발견된 시스템의 service rate의 특성을 이용하여 service rate를 측정한 후 M/M/s 시스템의 balance of state equation

$$\lambda(n)\,Q(n)=\mu\,(n+1)\,Q\,(n+1)\cdots\cdots\cdots(1)$$

을 이용하여 steady-state probability Q(n)을 구한다. 여기서 $\lambda(n)$과 $\mu(n)$ $(n=0,1,2,\cdots)$은 각각 시스템에 n 명이 있을 때의 arrival rate 와 시스템의 service rate를 표시한다. 본 시스템에서는 $\lambda(n)=\lambda$, $n=0,1,2\cdots$, 이고 서어버 각자의 service rate는 μ로 주어진다. 본 연구는 세 부분으로 이루어 지는데 먼저 traffic intensity $\hat{\rho}$를 계산하는 방법에 이어 $\hat{\rho}$가 아주 낮은 경우와 $\hat{\rho}$가 대략 0.8보다 낮은 경우의 개산법이 다루어진다. Very high traffic intensity 또는 non-stationary의 경우는 앞으로의 연구과제로 남겨진다.

기호 및 Traffic intensity의 개산

다수의 Server를 원하는 대기행렬의 state들은 $\{(n,m),\ n=0,1,2,\cdots,1\leqq m\leqq K^f\}$ 형태로 표시되며, 여기서 m은 system에 n 명의 고객이 있을 때 이 고객들의 type의 조합이 모든 가능한 조합중 m번째 조합을 이루고 있음을 의미한다.*

여기서 $f=\min(n,U)$.

이러한 state들로 이루어진 system의 변화과정을 Markov chain으로 나타내고자 할 때 T. P. M. T의 형태는 아래와 같이 주어진다.

$$T=\begin{bmatrix} 0 & A_0 & & & & & \\ B_1 & 0 & A_1 & & & & \\ & B_2 & 0 & A_2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & B_{U-1} & 0 & A_{U-1} & \\ & & & & B_U & 0 & W \\ & & & & & M & 0 & W \\ & & & & & & \ddots & \ddots & \ddots \end{bmatrix} \cdots\cdots(2)$$

상기의 행렬식에서

A_n; $K^n\times(K+1)^n$ upward jump probability를 나타내는 소행렬로서 state (n,m)에서 state $(n+1,m')$로 갈 확률들을 나타낸다.

B_n; $K^n\times(K-1)^n$ 행렬로서 state (n,m)에서 state $(n-1,m')$로 갈 확률을 나타낸다.

M, W; $K^U\times K^U$행렬로 $n\geq U$ 일때의 transition probability (downward, upward)를 나타낸다.

여기서 A_n들은 band diagonal, W는 diagonal matrix를 이룬다.

여기서 $R=W+R^2W$를 만족시키는 $K^U\times K^U$ rate matrix R을 정의하면 이 시스템에서 행렬 $D=M+W$는 irreducible stochastic 행렬이고 spectral radius $sp(D)=1$이므로 Neuts 〔3〕의 증명에 의해 행렬 R이 존재한다. 본 시스템에서의 equilibrium condition은 이 때 $sp(R)<1$로 주어지며 traffic intensity는

$$\rho=\frac{dWe}{dMe}\quad\text{로 정의된다.}\qquad〔1〕$$

여기서 $\underline{d}$는 $\underline{d}=\underline{D}\underline{d}$를 만족시키는 Vector이고 $e=(1,1,1,\cdots,1)'$이다.

따라서 정확한 ρ값을 구하기 위해서 $\underline{d}$ 및 W, M의 값이 알려져야 하는데 이들의 order는 K^U이다. 이 행렬의 order가 개산방법을 고안하게 된 주 원인이므로 여기서 행렬들의 크기에 관계

* 더욱 자세한 사항은 〔1〕의 page (2)를 참조할 것.

없이 계산이 가능한 ρ의 개산법의 필요성이 대두하게 된다.

이를 위하여 우리는 다음과 같은 방법을 사용하였다.

이제

$$\beta = \sum_{k=1}^{K} P_k d_k$$

라 하자. 여기서 P_k 는 한 고객이 type k 일 확률이다.

그러면 β는 고객들이 요구하는 server 수의 가중평균치 (weighted average) 가 된다.

또

$$\gamma = \frac{s}{\beta}$$

라 정의하면 우리는 traffic intensity 의 개산치로

$$\hat{\rho} = \frac{\lambda}{\gamma \mu}$$

를 사용한다.

이렇게 $\hat{\rho}$를 구하는 이유는 다음과 같다.

보통의 M/M/s 대기행렬에서 traffic intensity는 $\lambda/s\mu$로 주어진다. 이는 도착율과 최고 서어비스율 (서어비스가 동시에 고객을 처리할 수 있는 능력) 의 비율로 해석될 수 있으며 최고 서어비스율은 시스템 내에 s 명 이상이 있을 때 이루어 진다. 따라서 본 시스템의 경우 보통의 M/M/s 시스템에서의 s 와 같은 역할을 하는 량 (quantity) 를 찾아야 하며 γ를 그 량으로 선정했다.

그러나 본 시스템에서는 대기열에서 고객이 기다릴 경우에도 server들이 대기열의 제일 앞에 있는 고객의 수요를 충족시키지 못할 경우 놀고 있는 server들이 있을 수 있으므로 서어비스가 동시에 고객을 처리할 수 있는 능력은 β보다 적게 되며 따라서 γ가 보통 M/M/s 시스템의 s 역할을 하기에는 값이 커 $\hat{\rho}$는 당연히 ρ의 underestimation이 된다.

200개 이상의 모델에 상기 방법을 적용해 본 결과 $\hat{\rho}$는 항상 참값의 85~95% 범위에 들어 있었으며, 이는 뒤에서 소개될 Q(n) 의 개산방법에 기준치를 제공하는데 충분히 정확한 값임이 경험에 의하여 증명되었다.

Lower Traffic Intensity Case

여기서는 $\hat{\rho}$가 대략 0.3보다 적은 경우를 다루도록 한다.

이제 만일 시스템내에 server의 수가 무한히 있다면 모든 고객은 그가 몇명의 server를 요구하면 blocking 이 일어나지 않으므로 도착 즉시 서어비스를 받게 된다. 따라서 이 경우 보통의 M/M/∞ 시스템과는 서어비스를 하고 있는 server의 수가 다를 뿐이며 이는 steady-state probability Q(n)을 계산하는데 전혀 영향을 주지 않는다.

따라서 Q(n) 을 구하는데 관계된 한 이 경우 다수의 server를 요구하는 시스템은 단순히 보통의 M/M/∞ 시스템으로 간주할 수 있다.

M/M/∞의 경우 시스템내에 n 명이 있을 확률은 Poisson 분포를 따르게 되어

$$Q(n) = \frac{e^{-\rho}\rho'^{n}}{n!} \quad \cdots\cdots (3)$$

로 주어진다. 여기서 $\rho' = \frac{\lambda}{\mu}$

우리의 경우 만일 server의 수가 무한대이면 위의 이유들로 식(3)은 성립한다. 더구나 blocking이 일어나지 않으므로 Pr〔서어비스내의 한 고객이 type k〕$= P_k$. 따라서 시스템내에 n명이 있고 그들의 type이 $\ell_1, \ell_2, \cdots \ell_n$ 일 확률 $q^{(n)}_{\ell_1, \ell_2, \cdots, \ell_n}$은 단순히

$$q^{(n)}_{\ell_1, \ell_2, \cdots \ell_n} = Q(n) P_{\ell_1} \cdot P_{\ell_2} \cdots\cdots P_{\ell_n} \quad \cdots (4)$$

으로 주어진다.

Server의 수가 무한대가 아닐 경우라도 traffic intensity가 낮으면 도착하는 한 고객이 block당할 확률은 작다. 만일 이 block당할 확률을 무시한다면 (3)식이 성립된다. $\hat{\rho}$가 0.3보다 적을 경우 block당할 확률은 매우 작으므로 (3)식으로 Q(n)이 개산된다. 정확도는 $\hat{\rho}$가 적을수록 좋아지며 numerical example이 표 〈1〉에 제시되었다.

이때 모든 measure는 M/M/∞의 경우를 따른다. 도표에서 볼 수 있듯이 이 방법으로는 $\hat{\rho}$가 대략 0.3 정도로 증가할때까지는 상당히 정확한 결과를 얻을 수 있으나 이 값을 초과하여 증가하면 정확도가 점점 떨어지므로 이 방법은 대략 $\hat{\rho}$가 0.3정도까지는 안전하게 사용할 수 있다.

$s=5$, $K=2$, $d_1=2$, $d_2=2$ 의 경우

P_i : case 1 : $P_1=0.6$ $P_2=0.4$

case 2 : $P_1=0.6$ $P_2=0.4$

case 3 : $P_1=0.4$ $P_2=0.6$

〈표 1〉 exact / approximation

	λ	μ	ρ	ρ	Q(0)	Q(1)	Q(2)	Q(3)	Q(4)	Q(5)	TL
CASE 1	.7	1.0	.207	.196	.495 / .496	.347 / .348	.121 / .122	.029 / .028	.006 / .005	.001 / 0.001	.72 / .70
CASE 2	1.0	1.0	.296	. 28	.364 / .368	.364 / .184	.182 / .061	.062 / .061	.019 / .015	.006 / .003	1.008 / 1.0
CASE 3	1.0	1.0	.348	. 32	.358 / .368	.358 / .368	.179 / .184	.066 / .062	.024 / .015	.009 / .003	1.15 / 1.0

여기서 TL = Average total number of customers in the system.

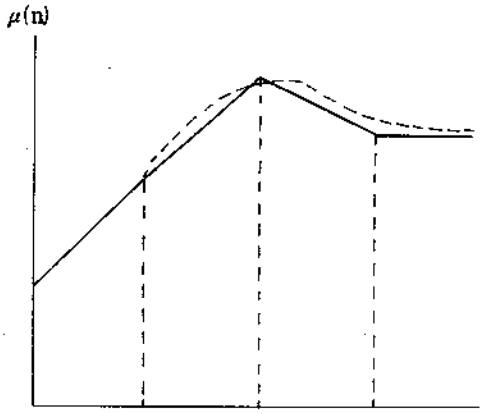
Medium or high traffic intensity case

〔1〕에서 연구되어 밝혀진 본 시스템의 서어비스율의 특성은 다음과 같다.

i) 최고의 서어비스율은 U보다 적고 이에 인접한 n에서 이루어지며,

ii) $\mu(n)$은 최고점에 도달한 후 서서히 감소하여 λ/θ $(=\lambda sp(R))^{-1}$로 수렴한다.

비록 $\mu(n)$은 step function이나 이러한 현상은 지속적인 형태를 띠우고 있으며 대략 그림 (1)의 점선 형태로 표시할 수 있다.



(그림 1)

여기서 채택한 개산법에서는 위와 같은 곡선의 형태를 점선으로 표시된 piecewise linear로 추정하여 $\mu(n)$ 값을 구하여 통상의 M/M/s system의 해법을 적용한다. 그림 (1)에서 n_1은 위에서 지적한 시스템의 특성이 (보통의 M/M/s와 다른) 나타나기 시작하는 점이고, n_2 는 $\mu(n)$이 최고값을 갖는 점이며 n_3는 $\mu(n)$이 λ/θ에 충분히 접근했다고 인정되는 점이다.

다음에서와 같이 n_1, n_2, 및 n_3를 구하고 이에 해당하는 $\mu(n_i)$들을 구한 후 각 점사이의 $\mu(n)$들은 각 n에 해당하는 $\mu(n)$ 값은 위 그림의 직선에서 찾는다.

i) n_1

n_1은 시스템내의 고객이 이들의 type에 관계없이 모두 서어비스를 받을 수 있을 때 시스템에 있을 수 있는 최고의 고객수이다. 즉 언제나 blocking이 일어나지 않는 고객수중 가장 큰수이다. 따라서 n_1보다 적은 n에서는 $\mu(n) = n$이 되고 n_1은

$$n_1 = \max_n \{nd_k \leqq s\} \quad \cdots\cdots (5)$$

로 정해진다.

ii) n_2

$\mu(n)$이 최고점에 달할 때의 고객수 n_2는 고객 type의 수, 종류 및 traffic intensity에 따라 다르다. 그러나 일반적으로 최고 서어비스 율은 U와 같거나 매우 근접한 점에서 일어나므로 우리는 n_2의 값을 U로 추정하였다. n_2에 해당하는 서어비스 rate를 얻기 위하여 앞에서 구한 평균 서어비스 요구수 β와 $\gamma = s/\beta$를 이용하였으며 $\mu(n_2) = \gamma\mu$로 추정한다. β는 순수한 수학적인 가중평균치이므로 blocking현상이 고려되어 있지 않기 때문에 $n = U$에서의 실제의 고객의 평균 Server 요구수는 β보다 크다. 따라서 $\gamma\mu$는 $\mu(n_2)$의 참값의 overestimation이 된다. 그러나 그림〔1〕에서 볼 수 있는 바와 같이 $\mu(n)$의 실제 증가율 (점선)은 그림의 직선처럼 빨리 감소하지 않으므로 n_1과 n_2 사이의 $\mu(n)$ 값들은 대부분이 underestimate 되므로 앞에서의 의도적인 overestimation과 서로 상쇄효과를 나타내며 이는 실제 경험에서 상쇄효과가 충분한 것이 증명되었다.

iii) n_3

Traffic intensity가 낮거나 중간 정도의 시

스템에서는 state probability는 적은 값의 n에서 큰 값을 가지며 n이 U를 넘어 증가할수록 이 확률은 급격히 감소하게 된다. 일반적으로 매우 적은 값의 확률을 갖는 state가 시스템의 measures of performance에 주는 영향은 미미하다. 다시말해 n≫U일 때의 state probability에 약간의 부정확성이 허용되어도 전체 시스템의 measure of performance는 매우 적은 영향을 받을 뿐이다. 이러한 확률들은 행렬 R의 dominant eigenvalue $\theta(=sp(R))$에 의하여 결정되며 〔2〕 따라서 우리는 θ값의 추정에 약간의 부정확성을 허용할 수가 있다.

여기서 먼저 행렬 R은 다음의 관계를 만족시킨다는 점을 상기하자

$$R = W + R^2 M$$

이제 $sp(R) < 1$이고 행렬 M의 각각의 열의 합(row sum)은 1보다 작고 양의 요소(positive entries)가 골고루 퍼져 있으므로 행렬 R^2M의 요소들은 크기가 매우 작으며 골고루 분포되어 있다. 반면 행렬 W는 diagonal행렬이며 이들의 요소(element)는 R^2M에 비교하여 크다(대부분의 경우는 상당히). 따라서 행렬 R의 diagonal entry는 다른 요소들 보다 크다. 우리의 경험으로는 예외없이 대략 ρ가 0.8이하일 경우 R은 diagonally dominant matrix였다. 만일 $R=(r_{ij})$이 완전한 diagonal행렬이라면 $\theta = \max_i r_{ij}$이다. 그러나 값이 작은 non-diagonal요소들이 존재하기 때문에 $\theta > \max_i r_{ij}$이며 일반적으로 $\max_i W_{ii} < \max_i r_{ii} < \theta$이다.

Traffic intensity가 증가함에 따라 R^2M의 요소들도 증가하며 따라서 $\max_i W_{ii}$와 θ의 차이는 커지게 된다. 이러한 사실들을 고려하여 우리는 θ의 추정치 $\hat{\theta}$를 다음과 같이 정한다.

$$\hat{\theta} = \delta \max_i W_{ii}$$

여기서

$$\delta = \begin{cases} 1.2 & \hat{\rho} \leqq 0.4 \\ 1.2 + (\hat{\rho} - 0.4) & \hat{\rho} > 0.4 \end{cases} \quad (6)$$

위와 같은 방법으로 δ의 값을 정하면 실제계산시 여기서 구해진 $\hat{\theta}$을 θ의 값으로 사용하는데 충분히 정확한 값임이 경험에서 밝혀졌다. W_{ii}들은 $\lambda/$(service를 받고 있는 고객수 $\cdot \mu + \lambda$)로 주어지므로

$$\max_i W_{ii} = \lambda / (\xi\mu + \lambda) \quad (7)$$

여기서 ξ는 s/d_K보다 적은 정수중 가장 큰 수.

$\mu(n)$이 λ/θ로 수렴하는 속도는 행렬 R의 order와 traffic intensity에 의하여 결정된다. 자연히 R이 커지고 ρ가 높아지면 수렴 속도는 저하한다. 우리의 경험에 의하면 중간의 traffic intensity를 갖는 경우 R의 order가 대략 50이 넘는 경우 60 step 이상이 지나야 $\mu(n)$이 λ/θ에 충분히 접근하였다. 또한 n>U일때 $\mu(n)$과 $\mu(n-1)$의 차이는 n이 U를 지난 후 처음의 몇 step에서 급격히 줄어든 후 서서히 줄어들며 0으로 접근한다. 이러한 사실들을 고려하면 n_3를 U의 연속형 함수로 추정할 수도 있으나 실제 계산시에 우리는 U+X의 형태로 추정하였으며 여기서 X의 값은 traffic intensity에 따라 변하며 이는 위에서 말한 현상-급격히 감소한 후 서서히 감소한-을 고려하여 경험치로 구해진다. 이러한 사실들을 종합하고 앞에서 언급한 200개의 실험모델에서 얻어진 경험에 의하여 n_3의 값을 다음과 같이 정하였던 바 실제 계산시 상당히 정확한 결과를 얻을 수 있었다.

$$n_3 = \begin{cases} U+5 & 0.3 < \hat{\rho} \leqq 0.4 \\ U+10 & 0.4 < \hat{\rho} \leqq 0.5 \\ U+15 & 0.5 < \hat{\rho} \leqq 0.6 \\ U+20 & 0.6 < \hat{\rho} \leqq 0.7 \\ U+30 & 0.7 < \hat{\rho} \leqq 0.8 \end{cases} \quad (8)$$

이 방법으로 n_3를 구하면 그림(1)의 n_3의 좌측에서 볼 수 있는 바와같이 $\mu(U)$의 overestimation이 보상된다. 물론 n_3에서의 $\mu(n)$은 λ/θ로 주어진다.

이상의 계산법은 다음과 같이 요약되어 진다.

step 0

(i) set $\beta = \sum_k d_k \rho_k$

(ii) set $\gamma = s/\beta$

(iii) set $\rho = \lambda/\gamma\mu$

(iv) 만일 $\rho \leqq 0.3$이면 lower traffic intensity case의 방법사용, 아니면 go to step 1

step 1

(i) $n_1 = \max_n [nd_K \leqq s]$

(ii) $\mu(i) = i\mu$, $1 \leqq i \leqq n_1$

step 2

(i) $n_2 = U$

(ii) $\mu(n_2) = \gamma\mu$

〈표 3〉

ρ	P_1	P_2	P_3	λ	μ	ρ	θ	Q(0)	Q(1)	Q(2)	Q(3)	Q(4)	Q(5)	L	Z
.316	.3	.4	.3	.6	1.	.257	.418 .45	.528 .53	.317 .318	.114 .102	.029 .033	.007 .010	.002 .003	.68 .682	.588 .6
.418	.3	.4	.3	.8	1.	.342	.572 .533	.416 .418	.333 .334	.142 .160	.060 .055	.026 .019	.011 .007	.968 .97	.722 .8
.518	.3	.4	.3	1.1	1.	.471	.649 .666	.272 .284	.30 .312	.176 .200	.10 .097	.058 .047	.035 .023	1.56 1.52	1.1 1.10
.627	.3	.4	.3	1.2	1.	.514	.693 .716	.229 .246	.276 .295	.177 .212	.109 .109	.069 .057	.045 .032	.190 1.81	1.342 1.2
.679	.3	.4	.3	1.3	1.	.557	.738 .767	.189 .210	.247 .273	.171 .213	.115 .118	.078 .068	.055 .040	2.42 2.19	1.456 1.3
.784	.3	.4	.3	1.5	1.	.642	.831 .865	.113 .131	.273 .196	.139 .177	.107 .113	.084 .075	.067 .050	1.21 5.59	1.562 1.50

2) s=7, K=3, $d_1=2$, $d_2=3$, $d_3=4$ 인 시스템의 계산결과

주1. 표(2)의 경우 정확한 결과를 얻기 위한 computer의 기억용량은 다음과 같다. server의 수는 s=5, 고객 type의 수는 K=2, 가장 적은 수의 server를 원하는 고객들이 요구하는 server의 수는 $d_1=1$이다. 따라서 이때 U=[5/1]=5, K=2이므로 T. P. M. T의 가장 큰 submatrix의 order는 $K^U=2^5=64$가 되어 사실상 이는 대형 computer외에서는 계산이 불가능한 크기이다. 실제 계산시 computer에서는 64×64 matrices M, W, R이 기억되어져야 하며 똑같은 size의 working matrix가 최소한 1개는 있어야 한다. 이 외에 이 보다 작은 matrix들 An, Bn(n=0,……, U)들도 역시 기억되어야 한다. 이러한 size의 경우 Cyber 72 computer로 100~140초 정도 계산시간이 걸린다.

결 론

서론에서도 언급하였듯이 본방법은 서어버의 수가 많거나 고객 type의 수가 많은 경우에 문제를 해결하기 위하여 고안 되었다. 그러나 이 방법은 각각의 서어버의 서어비스율이 다른 경우나 traffic intensity가 아주 높은 경우에는사용할 수 없다. 후자의 경우는 Diffusion process의 도입으로 전자의 경우는 적절한 방법으로 평균서어비스률을 구하는방법들이 고려될수 있으나 시스템의 복잡성으로 인하여 상당한 연구가 필요할 것으로 보여진다. 본 논문에 수록된 도표들은 비교적 시스템의 크기가 작은 경우인데 이는 정확한 결과와 비교하기 위하여 작은 시스템을 택한 것이다. 큰 시스템의 경우 Simulation의 결과들(정확한 결과라는 보장은 없지만)과 본 방법의 결과를 비교하였던 바 큰시스템의 경우에도 도표에서 보여준 정도의정확도가 유지됨이 밝혀졌다.

References

1. 김성식, 장진익, "Multi-Server Demanding M/M/s 대기행렬의 Service Rate 변화곡선에 관한 연구," 대한산업공학회지 Vol 6, No. 1 1980.
2. Kim S. S., "M/M/s Queueing System Where Customers Demand Multiple Server Use," ph. D. Dissertation, Department of Industrial Engineering, Southern Methodist University, 1979.
3. Neuts, M. F., "The Markov Renewal Branching Rocesses," Proc. Conf. on Math. Methods in the Theory of Queue, Kalamazoo, MI, Springer Verlag, N. Y. 1-21, 1974.

(iii) $\mu(i) = [(\gamma\mu - n_1\mu)/(U - n_1)] \cdot (i - n_1)$, $n_1 < i \leqq U$

step 3

(i) obtain n_3 from the equation (8)

(ii) $\xi = [s/d_K]$, max $W = \lambda/(\xi\mu + \lambda)$ where $[X]$ is the greatest integer less than X.

(iii) choose δ using the eguation (6)

(iv) $\theta = \delta$ max W

(v) $\mu(n_3) = \lambda/\theta$

(iv) $\mu(i) = (1/n_3 - U)(\gamma\mu - \lambda/\theta)(i - U)$, $U < i < n_3$

(iii) $\mu(i) = \lambda/\theta$, $i \geqq n_3$

위의 개산법에서 구해진 $\mu(n)$, $n = 0, 1, 2 \cdots$ 등을 사용하여 (1)식을 풀면 각각의 확률들 Q(n)은 다음과 같이 구해진다.

$$Q(o) = [1 + \Sigma_{i=1}^{n_1}(\lambda^i/\prod_{i=1}^{n}\mu(i)) + \lambda^{n_3}/\prod_{i=1}^{n}(1-\hat{\theta})^{-1}]^{-1}$$

$$Q(n) = \lambda^n/\prod_{i=1}^{n}\mu(i) \cdot Q(o), \quad 1 \leqq n \leqq n_3$$

$$Q(n) = Q(n_3)\,\hat{\theta}^{n-n_3} \qquad n > n_3$$

그리고 시스템내에 있는 평균고객수, L은 다음과 같다.

$$L = \Sigma_{n=1}^{\infty} nQ(n)$$

$$= \Sigma_{n=1}^{n_3-1} nQ(n) + Q(n_3)(n_3/(1-\theta) + \hat{\theta}/(1-\hat{\theta})^2)$$

또 $\mu(n)$들은 서어비스율이므로 어떤 특별한 n에서 서어비스내에 있는 평균 고객수, s(n),은 $\mu(n)/\mu$가 된다. 따라서 서어비스를 받고 있는 고객의 평균 숫자 Z는

$$Z = \Sigma_{n=1}^{n_3-1} s(n)\,Q(n) + \sum_{n=n_3}^{\infty} s(n_3)\,\theta^{n-n_3}$$

$$= \Sigma_{n=1}^{n_3-1} s(n)\,Q(n) + s(n_3)/(1-\theta)$$

따라서 대기열에서 기다리는 평균고객수, $\hat{L}$은

$$\hat{L} = L - Z$$

Little의 방법에 의하여 시스템내에서의 대기시간 W와 대기열에서의 대기시간 $\hat{W}$은 각각

$$W = \lambda L$$
$$\hat{W} = \lambda\hat{L}$$ 로 쉽게 계산된다.

아래의 도표에서는 전술한 계산방법에 의한 실제계산의 결과가 도시되었다.

〈표 2〉

exact /approximation

ρ	P_1	P_2	λ	μ	ρ	θ	Q(0)	Q(1)	Q(2)	Q(3)	Q(4)	Q(5)	L	Z
.278	.4	.6	.8	1.	.256	.345 .343	.444 .440	.355 .352	.142 .141	.042 .047	.012 .014	.004 .0035	.833 .858	.176 .800
.3776	.3	.7	1.	1.	.34	.40 .437	.354 .351	.354 .351	.177 .175	.069 .076	.027 .0789	.011 .009	1.20 1.131	.89 1.0
.486	.4	.6	1.4	1.	.448	.558 .514	.225 .222	.371 .311	.223 .218	.114 .129	.057 .065	.029 .0293	1.81 1.71	1.38 1.4
.589	.4	.6	1.7	1.	.544	.661 .6175	.150 .151	.259 .257	.222 .218	.138 .156	.084 .096	.052 .052	2.33 2.30	1.82 1.70
.734	.4	.6	1.7	.8	.680	.821 .762	.062 .079	.144 .168	.158 .179	.126 .160	.097 .123	.077 .084	3.627 3.748	2.10 2.125
.819	.4	.6	1.9	.8	.76	.983 .846	.002 .045	.017 .01	.017 .108	.016 .128	.016 .128	.015 .111	8.245 7.12	4.46 2.37

$s = 5$, $K = 2$, $d_1 = 1$, $d_2 = 2$ 인 시스템의 계산결과