韓國 컴퓨터情報學會 論文誌
第10卷 第4號, 2005. 9.

2005-10-4-1-24

큐잉 네트워크 모델을 적용한 저장용량 분석 시뮬레이션

김 용 수*

Simulation of Storage Capacity Analysis with Queuing Network Models

Yong-Soo Kim*

요 약

데이터 저장장치는 서버의 내부나 근처에 있는 것으로 인식되어 왔으나 네트워크 기술의 발달로 저장장치 시스템은 주 전산기와 원거리에 떨어져 존재할 수 있게 되었다. 인터넷 시대에 데이터 량의 폭발적인 증가는 데이터를 저장하는 시스템과 이를 전송하는 시스템의 균형 있는 발전을 요구하고 있으며 SAN(Storage Area Network)이나 NAS(Network Attached Storage)은 이러한 요구를 반영하고 있다. 저장장치로부터 최적의 성능을 도출하기위해서 복잡한 저장 네트워크의 용량과 한계를 파악하는 것이 중요하다. 파악된 데이터는 성능 조율의 기초가 되고 저장장치의 구매 시점을 결정하는데 사용될 수도 있다. 본 논문에서는 저장 네트워크 시스템의 큐잉 네트워크를 통한 분석적 모델을 제시한 다음, 이의 시뮬레이션하여 분석적 모델이 정당하다는 것을 입증한다.

Abstract

Data storage was thought to be inside of or next to server cases but advances in networking technology make the storage system to be located far away from the main computer. In Internet era with explosive data increases, balanced development of storage and transmission systems is required. SAN(Storage Area Network) and NAS(Network Attached Storage) reflect these requirements. It is important to know the capacity and limit of the complex storage network system to get the optimal performance from it. The capacity data is used for performance tuning and making purchasing decision of storage. This paper suggests an analytic model of storage network system as queuing network and proves the model though simulation model.

▸ Keyword : 저장장치, 큐잉 네트워크 모델, 용량, 분석적 모델, 시뮬레이션 모델(storage network, queuing network, capacity, analytic model, simulation Model)

• 제1저자 : 김용수
• 접수일 : 2005.07.13, 심사완료일 : 2005.09.05
* 경원대학교 소프트웨어대학 전자거래학부 교수

Ⅰ. 서 론

시스템의 용량은 현저한 성능저하 없이 증가하는 작업부하를 서비스할 수 있는 능력으로 정의할 수 있다. 용량, 작업부하, 성능 등은 모두 서로 연관되어 있고 서비스수준협약(SLA, Service Level Agreement)이나 성능 메트릭(metric)에 측정 가능한 수치로 표현되어야 한다[1][2]. 용량계획(Capacity Planning)은 주어진 시스템을 어떤 작업부하가 포화상태로 만드는지를 파악하여 시스템을 원하는 SLA를 만족시키게 하는 과정을 말하며 용량관리란 현재의 용량이 적절하며 가장 효율적인 방법으로 사용되고 있음을 확인하는 과정이다[3]. 저장 네트워크는 복잡한 시스템이기 때문에 용량계획은 쉽지 않고 잘못된 용량계획은 성능의 저하 또는 불필요한 비용의 낭비를 초래한다. 여러 가지 요인을 고려하여 저장시스템을 모델링하면 이를 이용해서 미래의 용량을 예측할 수 있는데, 주요 고려사항으로는 시스템의 구조, 하드웨어 및 소프트웨어 요소, 사용 패턴, 서비스 수준, 용량 등을 들 수 있다. 시스템을 이루고 있는 구성요소들 각각이 모두 필요한 용량을 만족시킬 수 있다고 하드라도 각 구성요소의 용량의 합이 전반적인 시스템 용량과 다를 수 있기 때문에 용량계획은 전체시스템-수준에서 이루어져야 한다.

데이터 저장장치는 서버의 내부나 근처에 있는 것으로 인식되어 왔으나 네트워크 기술의 발달로 저장 장치 시스템은 주 전산기와 원거리에 떨어져 존재할 수 있게 되었다. 인터넷 시대에 데이터 량의 폭발적인 증가는 데이터를 저장하는 시스템과 이를 전송하는 시스템의 균형 있는 발전을 요구하고 있으며 SAN(Storage Area Network)이나 NAS(Network Attached Storage)은 이러한 요구를 반영하고 있다. 전산시스템에 재해가 발생했을 때 저장장치에 들어있는 데이터의 복구가 가장 중요하며, 이러한 재해복구시스템의 문제점에 대한 논의도 제기되고 있다[4].

저장장치로부터 최적의 성능을 도출하기위해서 복잡한 저장 네트워크의 용량과 한계를 파악하는 것이 중요하다. 파악된 데이터는 성능 조율의 기초가 되고 저장장치의 구매시점을 결정하는데 사용될 수도 있다. 그런 의미에서 용량계획에는 주요 비용을 찾아내어 모델링하는 일이 포함되어야 한다[5].

저장 네트워크 시스템은 큐잉 네트워크로 모델링할 수 있다. 본 논문의 II장에서는 저장 네트워크 시스템의 분석적 모델을 만들기 위한 이론적 배경에 대해 논의하고, III장에서 분석적 모델을 제시한 다음, IV장에서 시뮬레이션 모델을 통해 분석적 모델이 정당하다는 것을 입증한 후 V장에서 결론을 맺는다. 이 모델은 입력 및 상황변수를 여러 가지로 변화시킴으로써 새로운 환경에서의 용량을 예측하는데 사용할 수 있다.

Ⅱ. 배경

큐잉 네트워크는 컴퓨터 시스템의 성능을 연구하는데 수십 년 동안 성공적으로 사용되어 왔다[6]. 큐잉 네트워크는 시스템을 서비스 센터와 연관된 큐의 네트워크로 간주하며 각 큐의 행위가 큐잉 이론의 핵심이 된다[7]. 분석적 모델은 시뮬레이션 모델이나 실험 분석보다 구성하기가 쉽고 개발시간이 덜 걸리며 빨리 결과를 도출할 수 있으나 그 결과가 상대적으로 부정확하다. 그러나 분석적 모델로부터 얻은 시스템 성능의 분석 결과는 시스템을 이해하는데 대단히 중요하다. 분석적 모델은 시뮬레이션 모델이나 실험 결과의 타당성을 증명하는 데 사용될 수 있다. 저장 네트워크 시스템의 분석적 모델을 구축하기 위해 필요한 큐잉에 관한 법칙을 재고해 볼 필요가 있다. 기본적인 성능 메트릭은 시스템에 도달하는 트랜잭션과 떠나는 트랜잭션을 관찰함으로서 얻을 수 있다[8]. 이러한 기본적인 성능 메트릭으로부터 큐잉 네트워크를 분석하는데 필요한 여러 가지 법칙을 유도할 수 있다.

$$\lambda = A/T \quad \cdots\cdots (1)$$

(λ: 도착율, A: 도착 트랜잭션 수, T: 관측 시간)

$$X = C/T \quad \cdots\cdots (2)$$

(X: 처리율, C: 종료 트랜잭션 수, T: 관측 시간)

안정된 상태(steady state)에서는 X = λ가 된다.

가장 중요한 성능 법칙은 Little의 법칙이며 다음과 같이

표현되며 트랜잭션 수가 보존되는 안정된 상태에 있는 어떠한 시스템에서도 적용된다[9].

$$N = X \times R \quad \cdots\cdots (3)$$

(N: 활동 중인 트랜잭션 수, R:응답시간)

안정된 상태에서 식 (3)은 식 (4)와 같이 표현될 수 있다.

$$Q = \lambda \times R \quad \cdots\cdots (4)$$

(Q: 현재 서비스 중인 것도 포함한 큐의 길이)

이용률을 시스템이 바쁜(busy) 시간을 총 관측시간으로 나누어 정의하고, 서비스 시간을 트랜잭션 하나를 처리하는데 걸리는 시간으로 정의하면 다음의 유명한 이용률 법칙(Utilization Law)을 얻을 수 있다.

$$U = B/T = (B/C) \times (C/T) = S \times X \quad \cdots\cdots (5)$$

(U: 이용률, B: 바쁜 시간, S: 서비스 시간)

큐잉 네트워크에서 한 트랜잭션이 어떤 서비스 센터를 여러 번 거쳐야 할 경우가 있다. Vi를 각 트랜잭션이 서비스 센터 i를 방문해야 할 수라고 가정하면 식 (6)과 같은 Forced Flow Law가 성립된다.

$$Xi = Vi \times X \text{ 또는 } Vi = Xi/X \quad \cdots\cdots (6)$$

(Xi : i의 처리율, Vi : i 방문 수)

한 트랜잭션이 시스템을 떠나기 전에 서비스 센터 i에 요구하는 서비스 요구(service demand, Di)는 Vi와 i에서의 평균 서비스 시간 Si를 곱해서 얻을 수 있고 이를 이용하여 식 (7)과 같은 Service Demand Law하고 한다.

$$Di = Vi \times Si = (Xi/X) \times (Ui/Xi) = Ui/X \cdot (7)$$

단일 큐잉 센터에서 응답시간은 서비스 시간과 큐에서 기다리는 시간의 합이다.

$$R = S + W \quad \cdots\cdots (8)$$

(W: 큐에서 기다리는 시간)

M/M/1 큐와 FIFO 큐를 가정할 경우, 한 트랜잭션이 도착했을 때 큐에 있는 트랜잭션의 수가 평균 Q이면 $W = Q \times S$로서 나타낼 수 있고 이를 식 (8)에 대입하여 정리하면 식 (9)을 얻을 수 있다.

$$R = S + W = S + (Q \times S)$$
$$= S + (X \times R \times S) = S + (U \times R)$$

그러므로 $R = S/(1 - (X \times S))$ 또는

$$R = S/(1 - U) \quad \cdots\cdots (9)$$

식 (9)의 양변에 X를 곱하고 Little의 법칙을 적용하면 식 (10)을 얻을 수 있다.

$$X \times R = (X \times S)/(1-U)$$
$$Q = U/(1 - U) \quad \cdots\cdots (10)$$

이 논문에서는 트랜잭션이 외부로부터 무한히 시스템에 도착한다는 가정을 하는 열린(open) 큐잉 네트워크 모델을 적용할 것이므로 열린 큐잉 모델만 살펴보기로 한다. 식 (6)과 (9)를 사용하여 식 (11)을 얻을 수 있다.

$$Ri = Si/(1 - X \times Vi \times Si) \quad \cdots\cdots (11)$$

큐잉 네트워크 시스템의 응답시간은 개별적인 큐에서의 응답시간의 합과 같으므로 식 (12)와 같은 일반적인 응답시간 법칙을 얻을 수 있다.

$$R = \Sigma i \ (Vi \times Ri) \quad \cdots\cdots (12)$$

각 센터의 이용률은 식 (1)과 (3)에 의해 식 (13)과 같이 처리율을 통해 나타낼 수 있다.

$$Ui = Xi \times Si = X \times Vi \times Si = X \times Di \cdots (13)$$

마지막으로 시스템의 큐 길이는 개별 큐의 길이의 합과 같다.

$$Q = \Sigma i\ Qi \cdots (14)$$

(Qi : 센터 i에서의 큐의 길이)

저장 네트워크 시스템에 대한 분석적 또는 시뮬레이션 모델이 만들어 지면 매개변수를 변경하여 새로운 환경에서의 시스템 동작을 예측할 수 있다. 이 논문에서는 용량분석 후 매개변수를 변화시켜 모델로부터 용량한계를 예측하려 한다.

III. 분석적 모델

열린 큐잉 네트워크에서, 임의의 서비스 노드에서 소비된 총 시간은 그 노드를 방문할 때 마다 걸린 응답시간을 모두 더한 것과 같다. 식 (11)은 M/M/1 서비스 노드에서의 방문 당 응답시간이다. 이 식의 좌 우변에 트랜잭션 당 방문 횟수를 곱하면 그 서비스 노드에서의 트랜잭션 당 총 응답시간을 얻을 수 있다.

$$Ti = Vi \times Ri$$
$$= Vi \times Si/(1 - X \times Vi \times Si) \cdots (15)$$

식 (7)의 서비스 요구에 대한 식을 식 (15)에 대입하여 식 (16)을 얻는다.

$$Ti = Di / (1 - X \times Di) \cdots (16)$$

식 (16)을 X에 대해 정리하면 식 (17)을 얻을 수 있다.

$$X = (1/Di) - (1/Ti) \cdots (17)$$

식 (17)은 서비스 노드 i에서 응답시간이 Ti를 야기한 시스템 전체의 처리율을 계산한 것이다. 식 (17)로부터 노드 i의 응답시간 Ti가 어떤 값 T'를 넘지 않을 때의 최대 처리율 X를 식 (18)과 같이 계산할 수 있다.

$$Ti \le T' \longrightarrow X \le (1/Di) - (1/T') \cdots (18)$$

식 (18)은 한 노드에 대한 응답시간의 한계를 가정한 것이며 전체 시스템에 대한 응답시간을 고려할 때는 Ti 대신 식 (12)의 R을 사용해야 한다.

M/M/1 노드 i에서 식 (13)을 사용하여 이용률 Ui가 최대 이용률 U'보다 작아야 하는 조건을 만족시키는 식 (19)를 얻을 수 있고, 이용률을 U'이하로 하기 위해 처리율은 U'/Di보다 작아야 한다는 것을 타나낸다.

$$Ui \le U' \longrightarrow X \le U'/Di \cdots (19)$$

(그림 1)은 하나의 서버와 두개의 저장 서브시스템으로 구성된 시스템을 나타낸다.

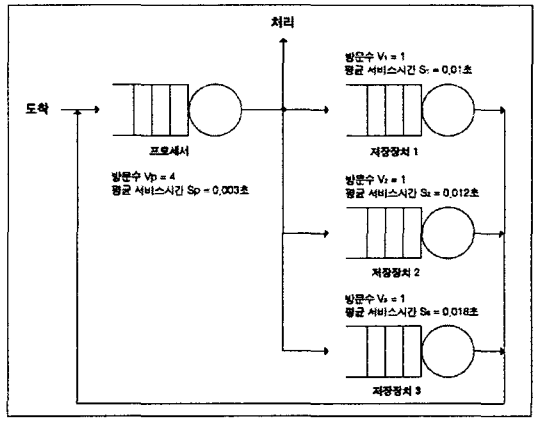

그림 1. 시스템의 구성
Fig. 1 System Structure

3.1 용량 분석

하나의 트랜잭션이 들어와서 모두 서비스 노드이며 M/M/1 큐인 프로세서와 저장장치를 (그림 1)과 같이 사용

한 후 시스템을 떠난다고 가정하자. 다음에 가정된 값들은 실제 운영 중에 추출한 데이터를 기초로 하였다.

프로세서 사용시간 Sp = 0.003,
저장장치 1 사용시간 S1 = 0.01,
저장장치 2 사용시간 S2 = 0.012,
저장장치 3 사용시간 S3 = 0.016
프로세서 방문 Vp = 4
저장장치 1 방문 V1 = 1
저장장치 2 방문 V2 = 1
저장장치 3 방문 V3 = 1

위의 조건에서 저장장치 서브시스템의 응답시간이 60ms 이하이고 이용률이 70% 이하로 하여 최대 도착율을 분석적 방법을 이용하여 구해보자.

서비스 요구는 다음과 같이 계산된다.

프로세서의 서비스 요구 Dp = 0.003 x 4
= 0.012
저장장치 1의 서비스 요구 D1 = 0.01
저장장치 2의 서비스 D2 = 0.012
저장장치 3의 서비스 D3 = 0.016

각 노드의 가능한 최대 처리율은 다음과 같다.

(식 20)
프로세서의 최대 처리율 1/Dp = 1/0.012 = 83
저장장치 1의 최대 처리율 1/D1 = 1/0.01 = 100
저장장치 2의 최대 처리율 1/D2 = 1/0.014 = 71
저장장치 3의 최대 처리율 1/D3 = 1/0.018 = 56

그러므로 저장장치 3 때문에 시스템의 처리율은 초당 56 트랜잭션을 넘지 못한다. 식 (18)을 이용하여 저장장치의 응답시간 60ms에 대응하는 처리율을 계산할 수 있다.

(식 21)
$X1 \le (1/D1)-(1/T1) = (1/0.01)-(1/0.06) = 83$
$X2 \le (1/D2)-(1/T3) = (1/0.014)-(1/0.06) = 55$
$X3 \le (1/D3)-(1/T3) = (1/0.018)-(1/0.06) = 39$

위 식에서 각 저장장치의 응답시간이 60ms를 넘기 전에 저장장치 1은 초당 83개, 저장장치 2는 55개, 저장장치 3은 39개의 트랜잭션을 처리할 수 있음 뜻한다. 또 식 (19)를 이용하여 이용률이 70%일 때의 처리율의 한계를 다음과 같이 계산 할 수 있다.

$X1 \le U'/D1 = 0.7/0.01 = 70$ ..... (22)
$X2 \le U'/D2 = 0.7/0.014 = 50$
$X3 \le U'/D3 = 0.7/0.018 = 39$

위 식에서 각 저장장치는 이용률이 70% 넘을 때까지 각각 초당 70개, 50개, 39개를 처리할 수 있다. 그러므로 50ms 이하의 응답시간과 70% 이하의 이용률을 원한다면 트랜잭션의 도착율은 초당 39개 이하이어야 한다.

3.2 한계 처리율

(그림 1)에서 각 노드의 이론적으로 가능한 최대 처리율은 식 (20)으로 나타낼 수 있다. 또 응답시간과 이용률에 제약을 설정하여 주어진 한계 내에서 최대 처리율을 식 (21)과 식 (22)를 이용하여 계산할 수 있었다. 식 (11), $Ri = Si/(1 - X \times Vi \times Si)$과 식 (12) $R = \Sigma i\ (Vi \times Ri)$를 이용하여 〈표 1〉과 같은 처리율과 응답시간과의 관계를 얻을 수 있다. 식 (20)에서 저장장치 3의 이론적 최대 처리율이 56이므로 처리율을 56이상 가정하여 계산할 필요가 없다. 지면 관계상 〈표 1〉에서 도착율, 4-34, 40-47, 51-56에 대한 데이터를 생략했다. 안정된 상태를 가정했을 때 도착율과 처리율은 같으므로 도착율을 처리율로 대치한다. 〈표 1〉에서 시스템의 응답시간을 100ms로 제한했을 때 허용할 수 있는 트랜잭션의 최대 도착율은 초당 38개이다. (그림 2)는 〈표 1〉을 액셀을 사용하여 그래프로 그린 것이다.

표 1. 도착율과 응답시간간의 관계
Table 1. Arrival Rate vs. Response Time

λ	Rp	R1	R2	R3	R
1	0.003036	0.010101	0.012146	0.01833	0.043613
2	0.003074	0.010204	0.012295	0.018672	0.044245
3	0.003112	0.010309	0.012448	0.019027	0.044897
35	0.005172	0.015385	0.02069	0.048649	0.089895
36	0.005282	0.015625	0.021127	0.051136	0.09317
37	0.005396	0.015873	0.021583	0.053892	0.096744
38	0.005515	0.016129	0.022059	0.056962	0.100665
39	0.005639	0.016393	0.022556	0.060403	0.104992
48	0.007075	0.019231	0.028302	0.132353	0.186961
49	0.007282	0.019608	0.029126	0.152542	0.208558
50	0.0075	0.02	0.03	0.18	0.2375

(λ: 도착율, Rp: 프로세서 응답시간,
R1, R2, R3: 각각 저장장치 1, 2, 3의 응답시간,
R = R1 + R2 + R3)

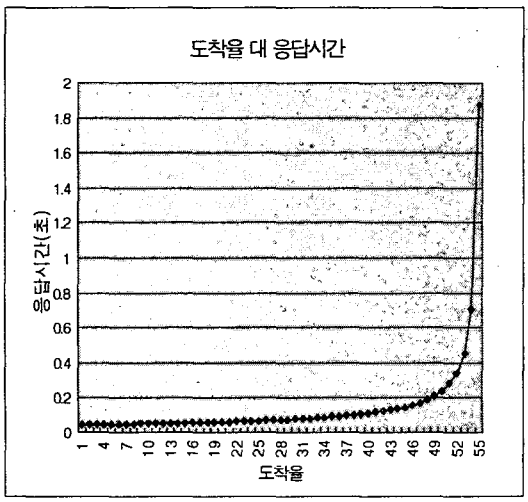

그림 2. 도착율 대 응답시간
Fig. 2 Arrival Rate vs. Response Time

식 (4) $Qi = \lambda i \times Ri$와 식 (14) $Q = \Sigma i\ Qi$를 이용하여 〈표 2〉와 같은 처리율과 큐의 길이간의 관계를 얻을 수 있다. 식 (20)에서 저장장치 3의 이론적 최대 처리율이 56이므로 처리율을 56이상 가정하여 계산할 필요가 없다. 지면 관계상 〈표 2〉에서도 도착율, 4-34, 40-47, 51-56에 대한 데이터를 생략했다. 〈표 2〉에서 시스템에서의 큐 길이를 10으로 제한했을 때 허용할 수 있는 트랜잭션의 최대 도착율은 초당 49개이다. (그림 3)은 〈표 2〉를 액셀을 사용하여 그래프로 그린 것이다.

표 2. 도착율과 큐의 길이와의 관계
Table 2. Arrival Rate vs. Queue Length

λ	Qp	Q1	Q2	Q3	Q
1	0.003036	0.010101	0.012146	0.01833	0.043613
2	0.006148	0.020408	0.02459	0.037344	0.08849
3	0.009336	0.030928	0.037344	0.057082	0.134691
35	0.181034	0.538462	0.724138	1.702703	3.146337
36	0.190141	0.5625	0.760563	1.840909	3.354113
37	0.19964	0.587302	0.798561	1.994012	3.579515
38	0.209559	0.612903	0.838235	2.164557	3.825254
39	0.219925	0.639344	0.879699	2.355705	4.094673
48	0.339623	0.923077	1.358491	6.352941	8.974131
49	0.356796	0.960784	1.427184	7.474576	10.21934
50	0.375	1	1.5	9	11.875

(λ: 도착율, Qp: 프로세서 큐의 길이,
Q1, Q2, Q3: 각각 저장장치 1, 2, 3의 큐의 길이,
Q = Q1 + Q2 + Q3)

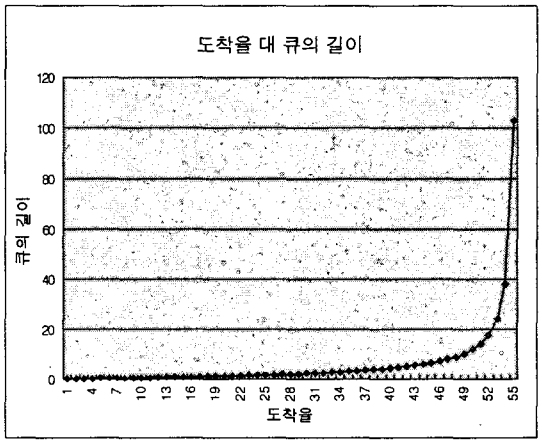

그림 3. 도착율 대 큐의 길이
Fig. 3 Arrival Rate vs. Queue Length

(그림 2)와 (그림 3)에서 보는 바와 같이 도착율이 50정도가 넘으면 응답시간과 큐의 길이가 급격히 커지는 것을 알 수 있는데 이 값을 문턱값(threshold)라고 하고, 용량한계를 추정하는 과정에서 시스템에 문턱값 이상의 도착율이 예상되면 시스템을 확장하든지 트래픽을 분산시켜야 한다.

Ⅳ. 시뮬레이션 모델

분석적 모델로 파악한 후 중요한 부분 또는 의심나는 부분을 시뮬레이션한다. 〈표 1〉에서 응답시간을 100ms 으로 했을 때 도착율을 38로 얻었고, 〈표 2〉에서 큐의 길이가 3.8일 때 역시 도착율을 38로 얻었다. 이를 역으로 도착율을 38로 가정해서 시뮬레이션해서 응답시간과 큐의 길이가 분석적 모델을 통해 얻은 것과 근사한지를 검증해보자. 같은 방법으로 도착율이 49일 때의 시뮬레이션도 해서 분석적 모델의 결과와 비교해 보자.

4.1 시뮬레이션 모델

시뮬레이션 모델은 SIMAN 언어를 기반으로 하고 GUI를 제공하는 범용적인 시뮬레이션 툴인 Arena를 사용하였다[10]. 시뮬레이션은 통계적으로 유효한 값을 얻기 위해 각각 20분씩 5번을 시행하여 평균값을 취하였다. (그림 4)는 (그림 1)을 Arena로 시뮬레이션한 모델이다.

프로세서의 서비스 시간은 분석적 모델에서는 산술평균이 가정되어있지만 시뮬레이션 모델에서는 프로세서의 특성상 중간값을 중심으로 10% 정도 상하로 최대값과 최소값을 갖는 삼각분포를 갖는다고 가정하자. 즉 최소값 0.0027초, 중간값 0.003초 그리고 최대값 0.0033초이다. 역시 저장장치도 분석적 모델에서는 서비스 시간이 산술평균이지만, 시뮬레이션 모델에서는 저장장치 1, 2, 3이 각각 평균0.01, 0.012, 0.018초인 지수분포를 갖는다고 가정하자. 초당 38개의 트랜잭션 도착을 가정하면 도착간 시간(inter-arrival time)이 0.026316초이며, 49개의 경우 0.020408초이다.

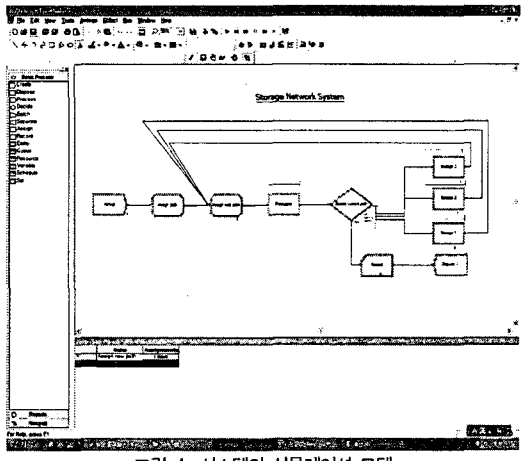

그림 4. 시스템의 시뮬레이션 모델
Fig. 4 Simulation Model of the System

〈표 3〉은 도착율이 38과 49로 해서 시뮬레이션한 결과의 응답시간과 큐 길이를 보여준다.

표 3. 시뮬레이션 결과
Table 3. Simulation Result

도착율	응답시간	큐 길이				
		Qp	Q1	Q2	Q3	계
38	0.1122	0.2177	0.2240	0.3699	1.4814	2.2930
49	0.2178	0.4778	0.4508	0.7989	6.4190	8.1465

3.2 분석적 모델과의 비교

〈표 4〉는 분석적 모델과 시뮬레이션 모델에서 추출한 응답시간에 대한 비교이며, 〈표 5〉는 큐의 길이에 대한 비교이다. 응답시간은 10% 이내의 차이를 보이지만 큐의 길이는 20-40%의 차이를 보이고 있다. 이는 분석적 모델이 산술평균을 사용하고 있지만 시뮬레이션 모델에서는 도착율과, 저장장치의 서비스 시간을 지수분포를, 프로세서가 삼각분포를 가진다는 가정을 했기 때문이다.

표 4. 응답시간 비교
Table 4. Comparison of Response Time

도착율	분석적 모델	시뮬레이션 모델	차이
38	0.100665	0.1122	10%
49	0.208558	0.2178	4%

표 5. 응답시간 비교
Table 5. Comparison of Queue Length

도착율	분석적 모델	시뮬레이션 모델	차이
38	3.825254	2.2930	40%
49	10.21934	8.1465	20%

실제 상황은 시뮬레이션 모델에 가까우므로 분석적 모델의 결과보다는 용량의 한계가 적을 것이다. 그러므로 시스템을 모델링할 경우, 먼저 시스템을 분석하여 분석적 모델을 만든 다음 실측치에 기준한 변수를 적용하여 결과를 얻고, 이를 시뮬레이션 모델을 통해 검정하며, 최종적으로 실제 시스템을 측정하여 검정한다. 이 세 과정은 서로 보완관계에 있으며 일관된 정보가 나올 때까지 반복해 가야한다.

V. 결론

인터넷 시대에 데이터 량의 폭발적인 증가는 데이터를 저장하는 시스템과 이를 전송하는 시스템의 균형 있는 발전을 요구하고 있다. 오늘날에는 네트워크의 속도가 저장장치 접근 속도보다 훨씬 빠르게 발전하고 있어서 저장장치에 네트워크 기술을 접목한 SAN이나 NAS가 널리 사용되고 있다. 인터넷으로 연결된 환경에서 사용자들에게 만족할만한 응답시간을 제공하기위해서는 이러한 복잡한 장치의 성능과 용량을 파악하는 것이 중요하다. 파악된 데이터는 성능 조율의 기초가 되고 저장장치의 구매 시점을 결정하는데 사용될 수도 있다. 본 논문에서는 저장장치 시스템을 큐잉 네트워크 모델로 만들어서 용량의 한계와 응답시간 및 큐의 길이와의 관계를 정립하고 이것을 다시 시뮬레이션 모델을 만들어 분석적 모델이 정당하다는 것을 입증하였다. 이 모델은 입력 및 상황변수를 여러 가지로 변화시킴으로써 새로운 환경에서의 용량을 예측하는데 사용 할 수 있다.

참고문헌

[1] 김용수, "정보체계 운영 아웃소싱에 있어서의 서비스 수준 측정 메트릭", 한국컴퓨터정보학회 논문지, 제9권 제2호, p69-79, 2004.6

[2] Yong-Soo Kim "AVAILABILITY METRICS OF AN ONLINE SYSTEM", Computer Measurement Group, Rino 1999

[3] Huseyin Simitchi, Storage Network Performance Analysis, WILEY, p199, 2003

[4] 김용수, 백승문, "국내 금융권 재해복구시스템의 문제점 분석", 한국컴퓨터정보학회 논문지, 제10권 제2호, p223-229, 2005.5

[5] J. Chase, D. Anderson, P. Thakar, and A. Vahdat, Managing Energy and Server Resources in Hosting Center, 18th ACM Symposium on Operating System Principles, pp103-116, Chateau Lake Louise, Banff, Canada

[6] Raj Jain, The Art of Computer Systems Performance Analysis, John Wileyy & Sons, April 1991

[7] Donald Gross and Carl M. Harris, Fundamentals of Queuing Theory, 3rd Edition, Wiley-Interscience, Dec. 1997

[8] P. J. Denning and J. P. Buzen, "The Operational Analysis of Queuing Network Models", Computing Surveys, Vol. 9-3, pp.223-252, Sep. 1977

[9] J.D.C. Little, "A Proof of the Queueing Formula L=λW", Operations Research, Vol.9, pp.383-387

[10] W. David Kelton, Randall P. Sadowski, David T. Sturrock, Simulation with Arena 3e, McGraw Hill, 2003

저 자 소 개

김 용 수

경원대학교 소프트웨어대학 전자거래학부

〈관심분야〉 운영체제, 성능분석/관리