韓國 컴퓨터情報學會 論文誌
第9卷 第3號, 2004. 9.

2004-9-3-1-6

2-계층과 3-계층 C/S 시스템의 응답시간 시뮬레이션

김 용 수*

Response Time Simulation for 2-tier and 3-tier C/S System

Yong-Soo Kim*

요 약

2-계층 C/S 시스템을 구성하는 경우 사용자의 응답시간이 길어지는 약점이 있다는 것이 기존의 C/S 시스템 개발자들에게는 잘 알려져 있으나 PDA 등을 이용하여 모바일 환경을 제공하는 새로운 개발자들은 곧 잘 이러한 사실을 모르고 종래의 2-계층 방식으로 시스템을 구성하여 실패하는 경우가 있다. 2-계층 C/S 시스템은 개발이 용이하다는 장점을 제외하고는 응답시간, 응용프로그램의 설치, 보안, 확정성, 응용프로그램의 재사용, 이기종 DBMS 지원, 가용성 등에 많은 문제가 있다. 본 논문에서는 간단한 트랜잭션이 2-계층과 3-계층 C/S 시스템에서 작동할 때의 응답시간을 시뮬레이션 하여 분석한다. 비록 문제공간을 2-계층과 3-계층에 국한해서 시뮬레이션 했지만 도출된 결과는 다계층 시스템에서도 적용할 수 있을 것이다.

Abstract

The fact that the end-user response time could become unacceptable in the 2-tier C/S system has been well known to C/S system developers. But inexperienced new developers who want to provide mobile solutions using PDA sometimes do not realize the ominous consequences of the 2-tier C/S systems. Except for the easy of development, the 2-tier C/S system has many weak points like in application deployment, security, scalability, application reuse, heterogeneous DBMS support, availability, and response time. This paper analyzes the response time in the 2-tier and 3-tier C/S systems through simulation. Although the problem space is confined to the 2 and 3 tiers. the result could be equally applied to any multi-tier systems.

▸ Keyword : 2-계층 시스템, 3-계층 시스템, 응답시간, 시뮬레이션

• 제1저자 : 김용수
• 접수일 : 2004.08.18, 심사완료일 : 2004.09.01
* 경원대학교 소프트웨어대학 소프트웨어학부

I. 서 론

컴퓨터 시스템 환경은 1990년대 초 매인프레임에서 클라이언트/서버(C/S, Client/Server)로 전환되는 다운사이징 열풍을 거쳐서 1990년대 말을 기점으로 웹 기반의 시스템으로 급속히 전환되고 있다. 매인프레임 환경에서 사용자는 터미널을 통해 문자 기반의 메시지를 매인프레임과 송수신하며, 시스템을 구현하는 로직은 모두 매인프레임에 존재했다. C/S 시스템은 클라이언트로 사용되는 급속히 향상된 PC와 서버의 성능을 효율적으로 이용하는 것 외에도 기업의 각 부서가 요구하는 고유한 서비스를 특성에 맞게 분산 지원케 하여 업무 효율성을 높이고 비용의 절감 효과를 추구하였다[1][2]. 초기의 2-계층 C/S 시스템에서는 GUI와 프로그래밍 로직을 가진 클라이언트가 DBMS 서버와 연결되도록 구성하였다. 이렇게 2-계층으로 구성한 시스템에서는 응답시간이 길어지는 문제가 발생하는데, 이러한 문제는 Tuxedo나 CICS와 같은 미들웨어(Middleware)를 포함하는 3-계층 시스템을 구성하거나 DBMS의 Stored-Procedure 등을 통해 해결할 수가 있었다. 분산시스템 환경에서는 CORBA (Common Object Request Broker Architecture) agent[3] 등을 통해 3-계층 이상의 다 계층 시스템이 구현되므로 여러 계층간의 정보전달이 사용자의 응답시간에 큰 영향을 줄 수 있다. 응답시간에 대한 2-계층 C/S 모델의 약점은 기존 개발자들에게는 잘 알려져 있으나 PDA 등을 이용하여 모바일 환경을 제공하려는 새로운 개발자들이 종래의 2-계층 방식으로 시스템을 구성하였다가 실패하는 경우가 있다. 2-계층 C/S 시스템은 개발용이성의 장점을 제외하고는 응답시간, 응용프로그램의 설치, 보안, 확정성, 응용프로그램의 재사용, 이기종 DBMS 지원, 가용성 등에 많은 문제가 있다[4]. 본 논문에서는 2-계층과 3-계층 C/S 시스템에 대해서 시스템 구성요소의 분산이 응답시간에 미치는 영향을 시뮬레이션 하여 분석한다. 응답시간은 사용자의 만족도에 가장 큰 영향을 미치는 성능 메트릭으로서 서비스 수준의 중요한 척도가 된다[5]. 비록 문제 공간을 2-계층과 3-계층에 국한해서 시뮬레이션 했지만 도출된 결과는 다계층 시스템에서도 적용할 수 있는 것으로 나타났다.

II. 시스템 구성

2.1 2-계층 C/S 환경

2-계층 C/S는 (그림 1)과 같이 클라이언트가 여러 개의 DB에 접근할 수 있게 되어 있으나 대개 하나의 DB 서버만 있고 LAN에서 운영되는 소규모 시스템이다. 성능과 유지보수의 편의성을 위해 Stored-Procedure를 이용해 DB에 한 번 접근하여 하나의 트랜잭션을 이루는 모양을 취하고 있다.

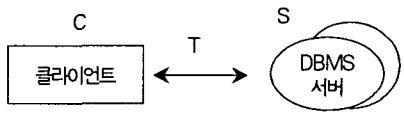

그림 1. 2-계층 C/S 구성도
Fig. 1 2-tier C/S Structure

클라이언트의 수행 시간을 C, 메시지 송수신 시간을 T, 서버에서의 수행시간을 S라 하면 사용자 응답시간은 식 (1)과 같이 상기 모든 시간을 합한 것과 같다.

응답 시간 = C + T + S ..... (식 1)

클라이언트의 요청과 서버의 응답은 다음과 같은 순서로 진행된다.

(1) 사용자의 요청
(2) 클라이언트 프로그램 수행
(3) 요청의 전송
(4) DBMS에서의 수행
(5) 응답의 수신
(6) 클라이언트 프로그램 수행
(7) 사용자의 응답 수신

(6)의 클라이언트 프로그램 수행 후에 (7)의 응답을 사용자 화면에 보여주고 끝날 수 있지만 다시 (3)의 DBMS 요청을 전송하여 (3)-(6)을 반복할 수도 있다.

하나의 트랜잭션을 이루는 경우는 다음 세가지 경우로 나눌 수 있다.

(가) (1)-(7) 수행하여 끝나는 경우
(나) (1)-(6)을 수행한 후 (3)-(6)을 1회 이상 반복한 후 (7)이 수행되어 끝나는 경우
(다) (1)-(7)을 1회 이상 반복하여 끝나는 경우

(가)는 간단한 경우이지만 (나)는 (3)과 (5)의 송수신 시간이 자신의 응답시간뿐만 아니라 다른 트랜잭션의 응답시간에도 큰 영향을 미칠 수 있다. (다)의 경우는 화면에 나타난 응답을 사용자가 본 후 다시 요청을 하는 형식이므로 사용자가 생각하는 시간이 다른 시간에 비해 너무 커서 다른 트랜잭션의 응답시간에 큰 영향을 미칠 수 있다. Stored-Procedure를 사용하는 2-계층 시스템은 (가)의 범주에 넣을 수 있다.

2.2 3-계층 C/S 환경

3-계층 C/S 환경은 (그림 2)와 같이 나타낼 수 있고 OLTP (Online Transaction Processing)는 트랜잭션 처리를 위한 미들웨어이지만 2-계층에 비해 응답시간을 향상시키는 역할을 하기도 한다.

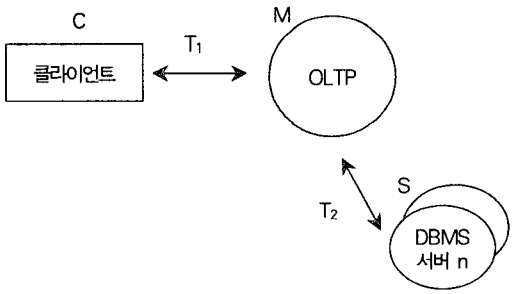

그림 2. 3-계층 C/S 구성도
Fig. 2 3-tier C/S Structure

클라이언트의 수행 시간을 C, 클라이언트와 OLTP간의 메시지 송수신 시간을 T_1, OLTP에서의 수행시간을 M, OLTP와 DBMS 서버간의 메시지 송수신 시간을 T_2, DBMS 서버에서의 수행시간을 S라 하면 사용자 응답시간은 식 (2)와 같이 상기의 모든 시간을 합한 것과 같다

$$\text{응답시간} = C + T_1 + M + T_2 + S \quad \cdots\cdots\cdots\cdots \text{(식 2)}$$

클라이언트의 요청과 서버의 응답은 다음과 같은 순서로 진행된다.

(1) 사용자의 요청
(2) 클라이언트 프로그램 수행
(3) 요청의 OLTP까지의 전송
(4) OLTP에서의 수행
(5) OLTP에서 서버까지의 전송
(6) DBMS에서의 수행
(7) 서버에서 OLTP까지의 전송
(8) OLTP에서의 수행
(9) OLTP에서 클라이언트까지의 응답 전송
(10) 클라이언트 프로그램 수행
(11) 사용자의 응답 수신

3-계층 시스템에서는 사용자 요청에 대해 (1)-(11)을 수행하면 하나의 트랜잭션이 끝나며 2-계층과 같은 여러 경우를 가정할 필요가 없다.

III. 시뮬레이션 모델

3.1 이론적 배경

사용자 응답시간은 (식 1)과 (식 2)에서 보는 바와 같이 프로세서를 사용하는 시간, DB에 접근하는 시간, 요청과 응답의 전송 시간 등으로 구성된다. 컴퓨터 자원은 여러 사용자가 공유하는 것이므로 자원이 어떤 요청으로 점유되어 있는 중에 그 자원을 요청한 사용자는 자원의 큐(queue)에 들어가고 자원이 사용가능 할 때 큐에서 꺼내어져 자원을 사용하게 된다. 그러므로 프로세서, 입출력, 메모리, 네트워크 자원 등 모든 컴퓨터 및 통신 자원은 (그림 3)과 같이 표현할 수 있는 큐를 가지고 있다.

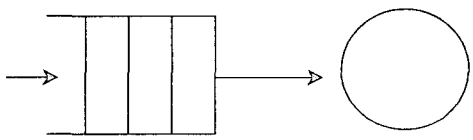

그림 3. 큐잉 시스템
Fig. 3 Queuing System

3.2 분석적 모델(Analytical Model)

3.2.1 프로세서

프로세스는 (그림 4)와 같은 상태전이 다이어그램과 같이 상태가 전이되고, Ready 큐에 있는 프로세스는 실제 프로세서를 사용하는 Running 상태로 전이되며 프로세서는 (그림 3)과 같이 수행된다[4].

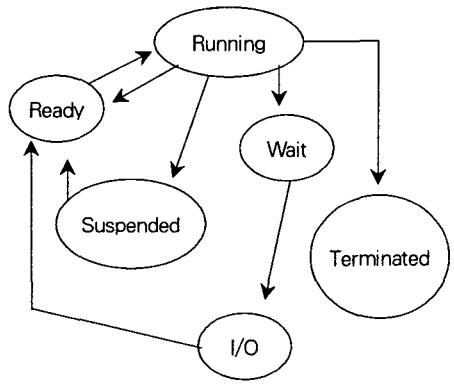

그림 4. 프로세스 상태 전이도
Fig. 4 Process State-Transition Diagram

프로세스의 전이를 따라 복잡하게 시뮬레이션 하는 것보다 (그림 3)과 같은 간단한 큐잉 모델로 CPU 사용에 대한 시뮬레이션을 하는 것이 효과적이다.

3.2.2 디스크 I/O

디스크 드라이브를 모델링하기 위해서는 다음과 같은 드라이브의 여러 가지 파라메터들, 즉 회전속도, 헤드 수, 표면 당 트랙 수, 평균 read/write seek 시간, 인접 트랙의 read/write seek 시간, Full-stroke write seek 시간, Internal formatted 전송률, Sustained 전송률, 캐쉬 버퍼의 크기, MTBF (mean time between failures) 등이 필요하다[5].

전형적인I/O 서브시스템에서의 I/O 순서는 다음과 같다[6].

(1) 응용프로그램으로부터 I/O 요청
(2) DBMS 또는 파일시스템 : 캐쉬에 저장한 후 디바이스 드라이버 구동
(3) 디바이스 드라이버 : 버스 어뎁터와 I/O 버스를 통해 디스크 컨트롤러에 전달한다. 컨트롤러가 바쁠 때는 큐에 넣는다.
(4) 디스크 컨트롤러 : 디스크 캐쉬에 저장한 후 디스크로 전달한다. 만약 디스크가 바쁘면 큐에 보관한다.
(5) 디스크 : ① Seek time, ②Rotational latency, ③ Head selection, ④ Read/Write 파라메타 들이 적용되어야 한다. Array 형태의 RAID (Redundant Arrays of Independent Disks)는 좀 더 복잡한 모형을 취하므로 본 논문에서는 단일 디스크의 경우만 논의 한다.

위의 순서 ④와 ⑤에 걸리는 시간의 평균을 S_a라 하면,

$$S_a = Controller\_Time + P_{miss}(Seek\_Time + Rotational\_Delay + Transfer\_Time) \quad \cdots\cdots\cdots \text{(식 3)}$$

여기서,
P_{miss} : 원하는 블록이 디스크 캐쉬에 없을 확률
Controller_Time : 컨트롤러에서 소요되는 시간(단위: 초)
Seek_Time : 평균 seek시간(단위: 초)
Rotational_Delay : 평균 회전시간(단위: 초)
Transfer_Time : 디스크와 컨트롤러 사이의 한 블록 전송 시간(단위 : 초)

Transfer_Time은 (식 4)와 같이 나타낼 수 있다.

$$Transfer\_Time = Block\_Size/(10^6 \times Transfer\_Rate) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \text{(식 4)}$$

여기서 Block_Size와 Transfer_Rate는 각각 디스크의 제조사가 제공하는 한 블록의 길이(단위 : 바이트)와 전송속도(단위 : 메가바이트/초)이다.

일련의 I/O를 수행하는 중에 연속된 블록을 액세스하는 것을 Run이라고 정의하면 하나의 Run에 대해 한 번의 Seek_Time이 소요된다. 한 블록의 액세스에 대해 Run을 고려해 정리하면 다음과 같다.

$$Pmiss = 1/Run\_Length \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \text{(식 5)}$$

$$Seek\_Time = Seek_a/Run\_Length \quad \cdots\cdots\cdots\cdots\cdots\cdots \text{(식 6)}$$

$$Rotational\_Delay = (1/2 + (Run\_Length - 1)((1+U_d)/2)) \times Disk\_Revolution\_Time/Run\_Length \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \text{(식 7)}$$

여기서 Run_Length: 한 Run의 연속된 블록의 수
U_d : 디스크 활용률(utilization)
$Seek_a$: 제조사가 제공하는 무작위 평균 seek 시간(단위: 초)
Disk_Revolution_Time : 디스크의 일 회전에 걸리는 시간(단위: 초)

Run이 1인 무작위(random) 액세스의 경우는 P_{miss}가 1인 되므로 평균 Seek_Time은 무작위 평균 Seek시간인 $Seek_a$와 같고 평균 회전시간은 (식 8)이 된다.

Rotational_Delay=Disk_Revolution_Time/2 ..... (식 8)

3.2.3 네트워크

여러 경로를 거치는 네트워크에서는 그 경로 중 가장 낮은 대역폭을 가지는 경로가 전체 지연시간을 지배한다. 본 논문에서는 단일 LAN 상에서 2-계층인 경우 같은대역폭을 갖는 네트워크를, 3-계층인 경우 클라이언트와 OLTP사이는 낮은 대역폭, 그리고 OLTP와 DBMS 서버 사이는 높은 대역폭을 갖는 스위칭 장비로 구성된 네트워크를 가정하였다. LAN은 가장 보편적인 Ethernet 및 TCP/IP로 구성되었다고 가정하였다. 하나의 메시지를 전송하기 위해 필요한 데이터그램의 수는 (식 9)와 같이 나타낼 수 있다.

#_Datagrams=Message_Size/MSS ..... (식 9)

여기서,
#_Datagrams : 데이터그램의 수
Message_Size : 전송 메시지의 크기 (단위 : 바이트)
MSS (Maximum Segment Size) : 최대 세그먼트 크기 (단위 : 바이트)

하나의 메시지를 전송하기위해 필요한 총 프로토콜 부하는 (식 10)과 같다.

OV=#_Datagrams x (TCPov+IPov+FRAMEov) ..... (식 10)

여기서,
OV : 총 프로토콜 부하 (단위 : 바이트)
TCPov : TCP 프로토콜 부하 (단위 : 바이트)
IPov : IP 프로토콜 부하 (단위 : 바이트)
FRAMEov : ethernet부하 (단위 : 바이트)

위 (식 9)와 (식 10)을 이용하여 전송시간을 (식 11)과 같이 나타낼 수 있다.

Transmission_Time=((Message_Size+OV)x8)/(10^6 xBandwidth) ..... (식 11)

네트워크에서 걸린 총 시간은 각스위칭 장비에서의 지연시간과 (식 11)의 전송시간을 합한 값이다.

3.3 시뮬레이션 도구의 선택

시뮬레이션 도구는 크게 세가지로 분류할 수 있다. 첫째, 중요한 기능을 라이브러리 형태로 제공하고 사용자는 그 제공된 기능을 이용하여 시뮬레이션 프로그램을 작성하는 경우이며 C++SIM, JavaSIM, baseSIM 등이 여기에 속한다. 둘째, 시뮬레이션 언어를 제공하여 사용자가 그 언어를 사용하여 시뮬레이션 하는 경우이며 SLAM II, GPSS/H 등이 여기에 속한다. 셋째, GUI를 사용하여 시뮬레이션의 개념과 대상만 알면 쉽게 시뮬레이션 할 수 있게 하는 것이다. Arena, Goldsim, SIMUL8, Extend 등이 여기에 속한다. 또한 네트워크, 분산운영체제 등 특정 기능을 시뮬레이션 할 수 있게 만든 시뮬레이션 툴들도 있는데 TCP/IP 네트워크를 위한 ns[9] 등이 여기에 속한다. 응답시간은 프로세서, 디스크 및 네트워크에서 걸린 시간을 모두 포함하기 때문에 네트워크 등 특정부분에 국한된 툴보다 범용적인 툴을 사용하는 것이 좋다. Arena[10]는 SIMAN 언어의 토대 위에GUI를 제공하는 범용적인 시뮬레이션 툴로서 많이 사용되고 있어서 본 논문에서 채택하였다.

3.4 환경 및 가정

시뮬레이션 모델을 만들기 전에 모델의 범위를 정하는 가정을 할 필요가 있다. 응답시간은 프로세서 사용, 디스크 접근, 네트워크 시간 등 여러 가지로 구성되어 있다. 이런 구성요소는 그 자체가 많은 시뮬레이션 대상이 되어 왔다. 본 논문에서는 분석적 방법으로 이러한 구성요소에 소요되는 시간을 계산한 다음 적절한 분포를 가정하여 시뮬레이션 모델을 작성하였다.

3.4.1 도착간격과 평균 큐 대기시간

각 구성요소에서의 트랜잭션 도착간격(μ_A)은 서비스시간

(μ_S)보다 커야 하며, 그렇지 않으면 트랜잭션이 쌓여가서 결국 포화상태가 될 것이다.

$$\mu_A > \mu_S \quad \cdots\cdots \text{(식 12)}$$

각 구성요소에서의 평균 대기시간은 트랜잭션의 도착간격과 서비스 시간의 관계로 (식 13)과 같이 나타낼 수 있다.

$$\text{평균 큐 대기시간} = \mu^2_S/(\mu_A - \mu_S) \quad \cdots\cdots \text{(식 13)}$$

3.4.2 호스트 소요 시간

대개 2GHz의 프로세서 사이클을 갖는 컴퓨터에서 각 구성요소에서 소요되는 전형적인 시간은 표 1과 같다[11].

표 1. 구성 요소 별 소요 시간
Table 1. Typical Times for Components

구성 요소	소요시간/액세스
프로세서	0.5ns (2GHz)
캐쉬 액세스	1ns (1GHz)
주기억장치 액세스	15ns
문맥교환	5,000ns
디스크 액세스	7,000,000ns
시간 할당량 (Quantum)	100,000,000ns (100ms)

3.4.2.1 프로세서 시간

Superscalar 기능이 있는 Intel의 Pentium chip은 1 사이클에 두 개의 명령어를 수행하도록 설계되었다. 그러므로 프로세서 사이클이 2GHz인 프로세서는 1초에 40억 개의 명령어를 수행한다. 어떤 기능을 10,000개의 명령어로 수행하고 수행 중에 한 번의 문맥교환이 있다고 가정하면 소요되는 시간은 10,000 x (0.5 / 2) x 1,000 + 5 x 2 + W = (12.5 + W)μs이 된다. 여기에서 2로 나눗셈한 것은 한 사이클에 2개의 명령어를 수행하기 때문이며 2를 곱한 것은 한 번의 문맥교환이 일어난 후 다시 문맥교환이 일어나야 수행을 계속할 수 있기 때문이다. 그리고 W는 문맥교환 후 Ready 큐에서 프로세서를 대기하는 시간이다. (식 14)는 상기 예를 일반화 한 것이다.

$$\text{P\_Time} = \#\text{\_Instructions}/(\text{I\_Time} \times 2) \times 1{,}000 + 2 \times \text{C\_Time} \times \text{C\_Num} + \Sigma W_i \quad \cdots\cdots \text{(식 14)}$$

여기서,

P_Time : 소요시간
#_Instructions : 기능을 수행하는데 필요한 명령어
I_Time : 명령어를 1개 수행하는데 걸리는 시간
C_Time : 문맥교환에 걸리는 시간
C_Num : 문맥교환 수
W_i : i번째 문맥교환 후 Ready 큐에서 대기하는 시간

3.4.2.2 디스크 액세스 시간

본 논문에서는 데이터베이스로부터 한 레코드를 읽거나 쓰는 것에 관심이 있으므로 임의의 한 블록을 액세스하는데 걸리는 시간을 논의하면 될 것이다. 한 제조사의 전형적인 디스크에 대해 컨트롤러 시간 0.1ms, 평균 Seek 시간 9ms, 분당 회전 수 7,200 RPM (Revolutions per Minutes), 전송률 20MB/초, 블록의 크기 2,048 바이트란 정보가 주어졌을 때 한 블록을 액세스하는데 걸리는 시간은 3.2.2에서 논의한 바와 같이 계산할 수 있다. (식 3), (식 4), (식 8)에 위의 정보를 대입하면 한 블록을 액세스하는데 걸리는 평균시간 13.37 ms을 얻을 수 있다. 만일 디스크 Arm이 해당 실린더에 이미 존재하여 Seek 시간이 없는 경우는 한 블록을4.37ms만에 읽어 들일 수 있다.

(식 3)으로부터,

$$S_a = \text{Controller\_Time} + P_{miss}(\text{Seek\_Time} + \text{Rotational\_Delay} + \text{Transfer\_Time}) = 0.1\text{ms} + 9\text{ms} + (1/2)\text{x}(1/(7200/60))\text{x}1000\text{ms} + (2048/(10^6\text{x}20))\text{x}1000\text{ms} = 0.1 + 9 + 4.17 + 0.1 = 13.37\text{ms}$$

3.4.3 네트워크 소요 시간

네트워크 소요시간은 각 장비에서 지연시간과 전송시간을 합한 것이며, 전송시간은 3.2.3의 (식 11)으로 구할 수 있다. 클라이언트와 DBMS 서버 및 클라이언트와 OLTP 서버와는 100Mbps의 대역폭을, OLTP와 DBMS 서버와는 1Gbps의 대역폭을 갖는 전형적인 시스템으로 가정하고 3-계층으로 구성할 때 100M/1Gbps 스위칭 장비에서의 지연을 0.2 μs, 메시지의 크기를 100 바이트, MSS를 1,460 바이트하고 가정하면 (식 9)에 의해 데이터그램 수는 1이 된다. 이러한 가정은 한 레코드의 정보를 수정하는 것을 시뮬레이션 하는 본 논문의 경우에 합당하다. TCP/IP와 Ethernet으로 구성된 네트워크를 가정하면 전형적인 부하는 (식 10)에 의해(1 x (20 + 20 + 18) = 58) 바이트가 된다. 각 대역폭에 대한 전송시간은 (식 11)에 의해 다음과 같이 계산할 수 있다.

1) 100바이트를 100Mbps의 대역폭으로 전송하는 경우
전송시간 = {((100 + 58) x 8) / (10^6 x 100)} x 10^6 = 12.64 μs
2) 100바이트를 1 Gbps의 대역폭으로 전송하는 경우
전송시간 = {((100 + 58) x 8) / (10^6 x 1,000)} x 10^6 = 1.264 μs

위의 전송시간에서 10M/1Gbps 스위칭 장비가 있는 경우 0.2 μs를 더하면 총 네트워크에 소요된 시간이 된다.

3.4.4 클라이언트 소요 시간

2-계층의 경우 첫 번째와 두 번째 SQL 요청 사이에 클라이언트에서 소요되는 시간을 가정해야 한다.

3.5 시뮬레이션을 위한 가정

3.4에서 논한 것을 바탕으로 시뮬레이션을 위한 여러 가지 변수들을 가정해야 하는데 본 논문에서의 가정을 요약하면 다음과 같다.

3.5.1 전송 속도

본 모델에서는 네트워크 프로토콜 지원 모듈이나 드라이버에서 소요되는 시간을 감안하여 100Mbps 대역폭 회선에서의 최단속도의 전송시간을 0.2 ms, 1Gbps에서는 0.02 ms으로 가정하였다. 속도는 하한값(0.2ms 및 0.02ms)을 기준으로 중앙값은 하한값의 1.2배, 상한값은 하한값의 1.5의 값을 갖는 삼각함수분포(Triangular Distribution)을 따른다고 가정 하였다. (Transmission Time = 0.2ms, High Transmission Time = 0.02ms)

3.5.2 문맥교환시간

3.4.2에서 논한 바와 같이 5μs로 가정하였다(CS=5μs).

3.5.3 DBMS에서 소요되는 시간

DBMS에서 소요되는 시간을 각각 8ms와 3ms로 가정하였다. (Pre CPU=8 ms, Post CPU=3 ms)

3.5.4 디스크 I/O

디스크로부터 읽어 들일 경우 80%는 버퍼에서 20%는 실제 I/O가 일어나는 것으로 가정하고 I/O는 하한 5ms, 중앙값이 하한의1.2배, 상한은 하한의 1.5배의 시간이 소요되는 삼각함수분포를 따르는 것으로 가정하였다. (IO=5ms)

3.5.5 Lock 시간

DBMS의 한 row를 수정할 때 적용되는 수정 lock의 시간 (Short Lock=10 μs)

3.5.6 사용자의 생각 시간 (Think Time)

사용자 시간도 하한 3초, 중앙값 9초, 상한 15초인 삼각함수분포를 따르는 것으로 가정하였다.

(Min Think Time=3초, Average Think Time=9초, Max Think Time=15초)

3.5.7 클라이언트 시간(2-계층에만 해당)

3.4.4.에서 논의한 클라이언트 시간(Client Time=200ms)

3.5.8 OLTP에서 소요되는 시간 (3-계층에만 해당)

(OLTP Time1=0.5ms, OLTP Time2=0.7ms)

3.5.9 트랜잭션 발생 간격

(식 12)과 같이 트랜잭션의 발생 간격시간은 전체 수행시간보다 커야 한다. 본 모델에서 초당 1건씩 트랜잭션을 발생시키는 경우 트랜잭션의 평균 수행시간이 2-계층의 경우 0.042초, 3-계층의 경우는 0.014초이므로 발생 간격으로 초당 1건을 택하였다.

Ⅳ. 시뮬레이션과 분석

4.1 시뮬레이션 모델

시뮬레이션 도구는 GUI를 사용하여 시뮬레이션 대상만 알면 쉽게 할 수 있는 것을 선택하였다.

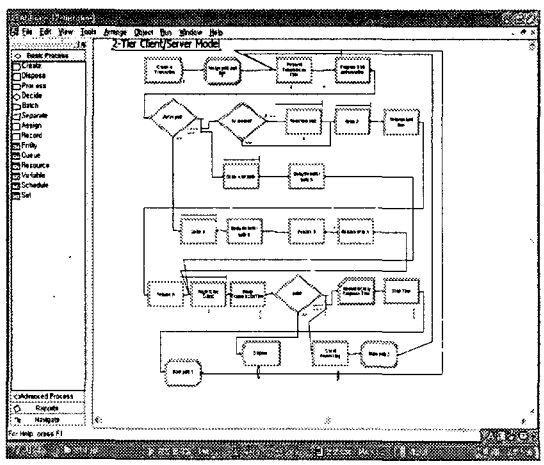

그림 5. 2-계층 C/S 시뮬레이션 모델
Fig. 5. 2-tier C/S Simulation Model

특정 부분에 국한된 것을 지양하고, SIMAN 언어를 기반으로 한 GUI를 제공하는 범용적인 시뮬레이션 툴인 Arena를 사용하였다. 2-계층 C/S 시스템을 위한 시뮬레이션 모델은 (그림 5)와 같다.

3-계층 C/S 시스템을 시뮬레이션 하기 위한 모델은 (그림 6)과 같다.

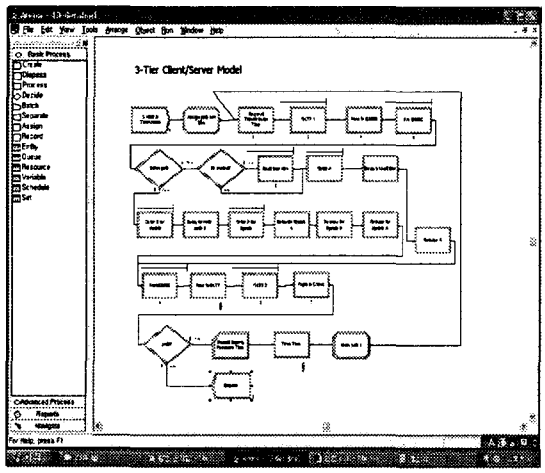

그림 6. 3-계층 C/S 시뮬레이션 모델
Fig. 6. 3-tirer C/S Simulation Model

4.2 분석

시뮬레이션은 제3장에서 논한 각 부분에 대한 데이터와 가정에 따라 30분씩 10번을 시행하였으며, 첫 2분간의 정보는 Warm-up 시간으로 제외하였다. (그림 5)의 2-계층 C/S 시스템과 (그림 6)의 3-계층 C/S 시스템에 대한 시뮬레이션 결과는 표 2와 같다.

표 2. 응답시간 (단위 : ms)
Table 2. Response Time (unit : ms)

구분	2-계층	3-계층	계층간 비율
평균	42.2	14.3	2.95
최소	11.4	12.7	0.90
최대	748.0	33.1	22.60
평균대최대 비율	17.7	2.3	-
최소대최대비율	65.6	2.6	-

〈표 2〉에서 보는 바와같이 응답 시간의 결과는 3-계층 C/S 시스템으로 구성할 경우 OLTP에서 소요되는 시간이 더해짐에도 불구하고 2-계층 C/S 시스템 보다 평균 응답시간은 약3배, 최대 응답시간은 약 23배나 차이가 나는 것을 알 수 있다. 최소 응답시간이 3-계층 C/S 시스템에서 길어진 것은 OLTP가 들어갔기 때문일 것이다. 평균 응답시간과 최대 응답시간의 비율은 3-계층에서 2.3인데 비해 2-계층에서는 17.7이며 최소 응답시간과 최대 응답시간의 비율도 3-계층에서는 2.6인데 2-계층에서는 65.6으로 3-계층으로 시스템을 구현했을 때 응답시간이 월등히 고르게 나타나는 것을 알 수 있다

V. 결론

종래의 2-계층 C/S 시스템은 응답시간, 응용 프로그램의 설치, 보안, 확장성, 응용 프로그램 재사용, 이기종 DBMS지원, 가용성 등에 문제가 있지만 개발이 용이하다는 장점으로 많이 사용되었다. 그러나 2-계층 구조에서 DBMS를 잘 못 사용하면 치명적인 성능 결함을 초래할 수 있다. 본 논문에서는 2-계층 구조에서 이러한 결함이 일어날 수 있다는 것과 3-계층 구조의 성능면에서의 우수함을 시뮬레이션 모델을 통해 제시하였다.

2-계층 구조의 문제는 Tuxedo나 CICS와 같은 미들웨어를 포함하는 3-계층 C/S 시스템을 구성하거나 DBMS의 Stored-Procedure 등으로 해결할 수 있다. 그리고 CORBA 등을 사용하여 3-계층 이상의 시스템을 구성하는 경우도 해당 컴포넌트 간의 데이터 전송속도 및 컴포넌트 내에서의 소요시간이 사용자의 응답시간에 지수함수적으로 영향을 미칠 수 있음을 2-계층 시스템으로부터 예측할 수 있다.

본 논문에서는 2-계층 C/S 시스템과 3-계층 C/S 시스템에 대해 GUI를 제공하는 범용적인 시뮬레이션 도구를 사용하여 (그림 5)와 (그림 6)의 모델로 시뮬레이션 한 결과 〈표 2〉와 같은 결과를 얻었다.

이 결과에 의하면 3-계층 C/S 시스템으로 구성할 경우 OLTP에서 소요되는 시간이 더해짐에도 불구하고 2-계층 C/S 시스템 보다 평균응답시간은 약3배, 최대 응답시간은 약 23배나 차이가 나는 것을 볼 수 있다. 이것은 3-계층 C/S 시스템에서 계층간 정보전달이 사용자 응답시간에서는 오히려 더 빠르다는 것을 증명하는 것이다. 따라서 클라이언트/서버 환경에서 개발자는 2-계층 C/S 시스템으로 구성할 경우 나타나는 약점을 잘 인식하여 가급적 피하고, 3-계

층 C/S 시스템으로 구성하는 것을 고려해야 할 것이다. 빠르게 변하는 모바일 개발환경에서도 2-계층 구조의 선택에 신중을 기해야 할 것이다.

그리고 본 모델은 3.5에서 가정한 변수들을 변화시켜서 여러 환경에 적합한 시뮬레이션으로 활용할 수 있다.

참고문헌

[1] Dan Trimmer, Downsizing Strategies for success in the Modern Computer World, Addison-Wesley, 1993

[2] Carl L. Hall, Technical Foundations of Client/Server Systems, John Wiley & Sons, 1994

[3] 장일동, "CORBA기반 지능형 에이전트 설계 및 구현", 한국컴퓨터정보학회 논문지, 제7권 제1호, 2002

[4] Robert Orfali, Dan Harkey, Jeri Edwards, Client/Server Survival Guide Third Edition, John Wiley & Sons, p.9, 1999

[5] 김용수, "정보체계 운영 아웃소싱에 있어서의 서비스 수준 측정 메트릭", 한국컴퓨터정보학회 논문지, 제9권 제2호, 2004, 6

[6] Jose M. Garrido, Performance Modeling of Operating Systems Using Object-Oriented Simulation, KA/PP, 2000

[7] Huseyin Simitchi, Storage Network performance Analysis, Wiley, 2003

[8] Daniel A. Menasce, Virglio A. F. Almeida, Capacity Planning for Web Services, Prentice Hall, 2002

[9] Richard Blum, Network Performance Open Source Toolkit, Wiley, 2003

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

[11] Kay A. Robbins, Steven Robbins, UNIX SYSTEMS Programming, Prentice Hall, 2003

저 자 소 개

김 용 수

경원대학교 소프트웨어대학 소프트웨어학부

〈관심분야〉 운영체제, 성능분석/관리