스트리밍 서버의 네트워크 스토리지 서브시스템 아키텍처에 관한 연구

전장환*, 정성훈**, 박춘식*, 이강우*
(b4crazy@dongguk.edu, shchung@sheenbang.com, chrispark@dongguk.edu, klee@dongguk.edu)

Research on the Architecture of Network Storage Subsystem in Streaming Servers

Jang Hwan Jun, Sung Hoon Chung, Chris Park, Kangwoo Lee

Abstract

급속한 정보 기술의 발전에 따라 인터넷 서비스의 고속화에 대한 요구가 급증하고 있다. 이에 따라 인터넷 관련 네트워크 장비들의 수요가 앞으로도 지속적으로 증대할 것으로 쉽게 예상할 수 있다. 하지만 국내 네트워크 관련 장비들은 아직도 외산에 많이 의존하는 실정이며, 특히 인터넷 망을 구성하는 핵심 요소인 인터넷 서버는 외산에 절대적으로 의존을 하고 있다. 따라서 개인용 컴퓨터 생산기술, 중대형 컴퓨터 설계 개발 기술, 메모리 설계 생산 기술 등 국내 산업계가 보유하고 있는 우수한 기술을 바탕으로 지속적인 연구 개발이 이루어져야 할 것이다. 본 논문은 인터넷 기반에서 1Gbps의 전송속도를 갖는 스트리밍 서버의 핵심 구성요소인 네트워크 스토리지 서브시스템의 구조를 제안한다. 본 연구를 통하여 국내에서도 중대형시스템을 위한 컴퓨터시스템의 구조에 대한 연구가 활성화되기를 기대한다.

Key Words: 인터넷 서버, 스트리밍 서비스, 네트워크 스토리지, 분포-구동 시뮬레이션, 컴퓨터 구조

* 동국대학교 정보통신공학과 (02-2260-8755, fax 02-2285-3343)
(서울시 중구 필동3가 동국대학교 정보문화관 Q동 204호)
** ㈜신방일렉트로닉스 부설연구소 (02-522-1175, fax 02-586-3746)
(서울시 서초구 서초동 1366-18 신방빌딩 3층)

1. 서 론

급속한 정보 기술의 발전에 따라 고성능 컴퓨터 시스템의 사용 범위가 인터넷 기반의 멀티미디어 서비스 서버에까지 확산되고 있다. 특히 많은 사용자를 동시에 수용하고 서비스를 안정적으로 제공할 수 있는 고성능·저 가격·고 가용성·확장성 등을 보유한 인터넷 스트리밍 서버 시장의 규모는 [표 1]과 같이 지속적으로 성장하고 있다.

[표 1] 국내·외 인터넷 서버 시장

	1999	2000	2001	2002	2003	2004
세계 시장	48,194	49,050	51,610	54,232	56,737	59,739
국내 시장	795	1,113	1,346	1,521	1,643	1,758

(단위: 백만불)

하지만 이러한 인터넷 스트리밍 서버 시장의 성정에도 불구하고 국내에서는 아직까지 독자적인 설계나 개발이 이루어지지 못하고 있다. 특히 시장 자체가 외산에 많이 의존하고 있으며, 컴퓨터 하드웨어 전공자 또한 3D 분야로 인식되면서 해마다 그 전공자가 감소하고 있는 실정이다. 이는 미국을 중심으로 중대형 서버를 설계, 생산하던 기업들이 인터넷 스트리밍 서버에 대한 연구와 개발을 10여년 전부터 지속적으로 이루고 있는 국외 상황과는 대조적인 현상이다. 이러한 상황을 극복하기 위해서는 국내의 컴퓨터 시스템 구조 연구 및 시스템 개발을 활성화하기 위한 다양한 연구가 필수적으로 선행되어야 한다.

따라서 본 연구에서는 기존 인터넷 스트리밍 서버 시스템의 성능을 개선하기 위한 대용량 저장 시스템과 입출력 기능을 강화, 스트리밍 서비스의 품질을 보장하기 위한 TOE의 설치, 네트워크 전용 프로세싱 유닛 설치 등의 새로운 시도를 통하여 인터넷 스트리밍 서버의 핵심 요소인 네트워크 스토리지 시스템(Network Storage Subsystem, NS)을 위한 최적의 구조를 제안한다.

본 논문의 제 2장에서는 본 논문의 주제와 관련된 연구들을 요약하고, 제 3장에서는 본 연구에서 구현하고자 하는 목표시스템을 소개한다. 제 4장에서는 분포-구동 시뮬레이션을 이용한 실험방법에 대하여 설명하고, 제 5장에서는 실험결과의 분석을 통하여 목표 시스템의 최적의 구조를 제안한다. 마지막으로 제 6장에서는 결론과 향후 연구 과제를 제시한다.

2. 관련 연구

[1]에서는 스트리밍 서비스를 위한 사용자의 요구 패턴을 분석하여 높은 효용도를 갖는 부하 균형을 유지하는 저장 시스템을 제안한다. [2]는, RAID의 개념을 일부 차용하여, RAIS(Redundant Array of Inexpensive Server)의 각 서버가 부하를 공유함으로써 부하를 줄이고 데이터의 저장과 출력을 늘리는 방안을 보여준다.

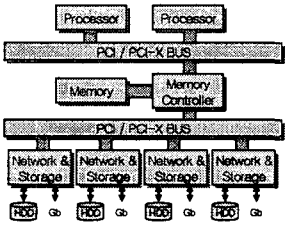
고성능 스트리밍 서버에의 병렬구조에 대한 연구는 매우 중요한 분야이다. [3]은 병렬구조의 서버의 동기화에 따르는 지연을 최소화하고 선형적인 성능 확장성을 확보하기 위한 스케줄링 알고리즘을 제안한다. [4]는 계층적으로 병렬화된 서버의 버퍼 관리 방법을 소개한다. 병렬 또는 분산 컴퓨팅의 성능특성을 비교 분석하기 위한 모델링 기법이 [5]에 소개되었다. 한편, 이와 같은 연구에 있어서는 시뮬레이션이 매우 유용한 기법이 되며[6], 또한 [7]에서는 정밀한 모델링을 위한 HDL 모델링 기법을 소개하였다.

[8]에서는 서버내의 효율적인 상호연결망의 구조로써 row-column crossover 버스와 확장성을 향상시키기 위하여 스위치를 장착한 분리형 버스를 제시하였다. [9]에서는 동적 객체 구조라는 새로운 개념을 제안한다. 이는 연속적인 미디어나 동기화, 동적 QoS 등을 위하여 객체지향 기술을 사용했다.

3. 목표 시스템

가. 시스템 개요

인터넷 기반의 스트리밍 서버는 고성능의 인터넷 기반 통신구조, 고품질의 무선통신환경, 많은 인터넷 사용량을 갖고 있는 인터넷기반 가상통신을 특성화시켜주는 시스템이다. [그림 1]은 스트리밍 서버의 하드웨어 구조를 보여준다. (소프트웨어적인 구조는 지면이 제한되어 있으므로 본 논문에서는 다루지 않기로 한다.) 스트리밍 서버는 일반적인 서버구조와 4개의 NS 등으로 구성되어 있으며, NS는 프로세서의 중재 없이 네트워크를 통해 zero-copy 방식으로 데이터를 전송할 수 있다.



[그림 1] 인터넷 스트리밍 서버의 구조

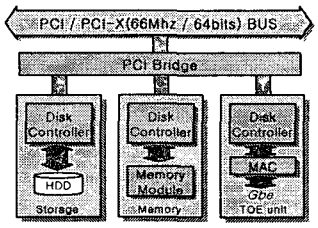
본 논문에서의 인터넷 스트리밍 서버 디자인 목표는 다음과 같다.

- 1,000명의 사용자를 지원, 200명의 사용자에게 동시에 4 Gbps의 전송을 지원
- 2Tbyte의 저장 공간으로 10 Gbyte의 200개의 HDTV와 같은 수준의 고화질 MPEG 파일 수용
- 서버는 서비스 요청 조절과 설정 세션의 시간 최소화를 통한 요구사항의 95%에 대하여 7초 이내에 스트리밍 서비스를 개시

나. 네트워크/스토리지 서브시스템

NS는 스트리밍 서버의 네트워킹과 저장된 데이터 사이의 처리를 책임진다. [그림 2]는 NS의 구조를 보여준다. NS를 구현하기 위한 하드웨어 블록도는 [그림 3]과 같다. NS 하드웨어는 PCI의 plug-in

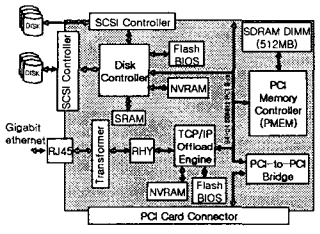
방식의 카드로 구현되어 스트리밍 서버의 PCI 슬롯에 장착된다.



[그림 2] NS의 구조

4. 시뮬레이션

본 논문에서는 스트리밍 서버의 NS를 분포-구동 시뮬레이션 패키지인 CSIM[10]을 이용하여 모델링한 시뮬레이터를 사용한다. 서버의 구조적인 특성상 부하의 분산이 원만하게 이루어졌다고 가정할 때, 4개의 NS는 모두 동일한 부하를 가지므로 우리는 하나의 NS가 동시에 50명의 사용자에게 스트리밍 서비스할 수 있다고 가정한다. 따라서 본 연구에서는 하나의 NS만을 고려하기로 한다.



[그림 3] NS의 하드웨어 블록도

가. 시뮬레이션 모델

하나의 스토리지 서브시스템을 구성하고 있는 요소들과 시뮬레이션 모델은 다음과 같다.

1) RAID 및 하드디스크

본 논문에서 고려하고 있는 Cheetah x15 하드디스크는 평균 48.9 MBps의 전송속도를 갖는데, 아래의 [표 2]는 하드디스크의 수와 블록의 크기에 따른 RAID의 데이터 전송속도를 나타낸다. [표 2]에 의거, 우리는 2MBytes의 블록 크기와 5개의 하드디스크로 RAID를 구성하며, 1Gbps(125MBps)의 대역폭을 충족시키기 위해서 2개의 RAID를 기본적인 시뮬레이션 모델로 책정하였다.

[표 2] 하드 디스크 수와 블록 크기에 따른 데이터 전송속도

디스크 수 / 블록크기	3	4	5	6	7
32 KB	3.3	3.5	3.0	3.0	2.9
128 KB	12	13	12	13	12
512 KB	39	45	46	47	48
2048 KB	80	94	104	118	123

(단위 : MBps)

2) 디스크 컨트롤러

디스크 컨트롤러는 Ultra SCSI160을 고려하여 Qlogic 12160 시스템을 모델로 정하였다. RAID의 수와 SCSI 컨트롤러의 수에 따라 정적으로 SCSI 컨트롤러를 배정하여 제어하도록 모델링 하였다.

3) PCI 및 Bridge

PCI 버스는 64 bits의 데이터 라인을 가지며 66 MHz의 속도로 작동한다. 중재 지연은 최대 7 버스 사이클로 하며, 장치들 간의 버스 사용권은 선입선출 방식으로 사용 허가를 부여한다.

4) 메모리 및 메모리 컨트롤러

NS의 메모리는 저장장치와 TOE 시스템사이에 위치하며 단순한 버퍼링 기능을 제공한다. 즉, 저장장치로부터 읽어 들인 데이터를 PCI 버스를 통하여 메모리에 저장하고, 이후에 TOE가 패킷을 구성하기 위하여 데이터를 필요로 할 때 메모리로부터 PCI 버스를 통하여 TOE 버퍼에 전송된다.

5) TOE

TOE는 목표 시스템에 가장 큰 영향을 미치는 요소로써, TOE 칩과 MAC 칩으로 구성되어 있다. TOE 칩은 스트림 패킷을 처리하며, PCI 메모리로부터 칩 내부의 버퍼로 데이터를 저장한다. MAC 칩은 병렬 구조 패킷을 직렬 구조 데이터로 풀어 전송하는 역할을 수행하지만 동작이 전체 시스템의 성능에 큰 영향 주지 않으므로 본 논문에서는 TOE 칩만을 고려하여 시뮬레이터를 구축하였다.

TOE는 처리하는 파일의 크기 단위에 따라 CPU를 점유율에 큰 변화를 주는데, 일반적으로 파일의 크기가 작아짐에 따라 CPU 점유율이 상승하는 추세가 뚜렷하며, 본 논문에서 채택한 Alacritech 1000x는 다음의 [표 3]과 같은 성능을 나타낸다.

[표 3] TOE의 패킷 처리 속도 (단위: Mbps)

파일 크기	2	4	8	16	32	64
처리 속도	470	800	1,210	1,520	1,620	1,740

(단위 : 파일 크기 KB, 처리 속도 Mbps)

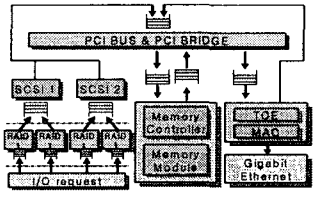
6) 요 약

최적의 구조를 찾기 위한 NS 각 구성요소의 성능 파라미터들은 다음의 [표 4]에 요약되어 있다.

[표 4] NS 구성요소의 사양 및 시뮬레이션 선택 사항

구성 요소	사양	시뮬레이션 옵션
RAID 및 하드디스크	△ Cheetah x15 (1,500 RPM) △ 2 MB 연속 읽기 보장 △ Seek time: 7.2 msec △ Head switch time: 1.5msec △ 전송속도: 104 MBps	RAID 수: 2, 4 개
디스크 컨트롤러	△ Ultra SCSI 160 지원 △ Qkogic 12160 △ 최소 1 Gbps 대역폭	디스크 컨트롤러 수 : 1, 2 개
PCI 및 Bridge	△ 64 bit 데이터 라인 △ 66 MHz △ Arbitration: 7 cycles △ 우선권 없음 △ 2MB 연속 전송	버스 수: 1, 2개
메모리 시스템	△ 용량 제한 (선택적) △ 접근 시간: 100 nsec	512M, 2G, 4G Bytes
TOE	△ Intel 82545EM △ 내부 버퍼: 16 MBytes △ 처리 속도: 470 ~ 1,740Mbps (파일의 크기에 따라 변화)	파일 크기: 1, 64 KB

나. 시뮬레이터



[그림 4] Queueing model

[그림 4]는 NS의 등가 대기열 모델이다. 본 대기열 모델에서 사용자의 요구를 처리하는 과정은 다음과 같

다. 즉, 사용자로부터의 서비스 요구에 의해 NS는 RAID로부터 데이터를 검출하여 디스크 컨트롤러와 PCI를 거쳐 NS 메모리에 적재한다. 이 데이터는 PCI를 거쳐 TOE를 통해 사용자에게 전송된다. 본 논문에서는 동시에 50개의 사용자 요구를 처리하도록 하고, 그 결과 분석을 통해 최적의 NS구성을 제시하였다.

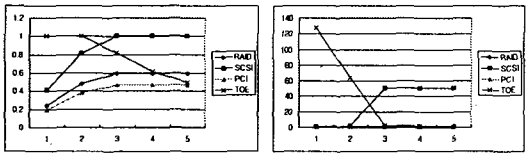
5. 시뮬레이션 결과 분석

가. 기본 시스템 사양 (TOE 파일 크기: 1KB)

우선 초기 시뮬레이션은 목표 시스템이 요구하는 사항을 배제한 채 NS의 가장 간단한 구조로 수행하도록 한다. 시뮬레이션의 결과 분석은 대기열 이론[11]에 의해, 모든 구성요소의 활용도가 30~50% 정도를 유지하도록 진행된다. [표 5]는 기본 시스템 사양의 각 구성요소들의 조건 표이며, 그 따른 시뮬레이션 결과는 [그림 5]와 같다.

[표6] 각 구성요소들의 조건 표

구성요소	조건
PCI 버스 수	1
PCI 버스 속도	66Mhz
RAID 수	2
SCSI 채널 수	1
SCSI 전송 속도	160MBps
TOE 수	1
TOE 파일 크기	1KBytes



(a) 평균 활용도 변화 그래프 (b) 평균 대기열 변화 그래프

[그림 21] 기본 사양에서의 평균 활용도 및 평균 대기열

[그림 5]를 보면 TOE의 수가 1~2개 일 때, TOE의 활용도는 90~100%, 대기열 길이는 130이다. 즉, TOE가 시스템의 병목이 되었다고 볼 수 있다. 이를 해소하기 위해 동일한 성능의 TOE를 여러 개 추가하는 방안과 TOE가 처리하는 파일의 크기를 증가시키는 방안을 사용할 것이며, 전자의 방법을 먼저 택하였다.

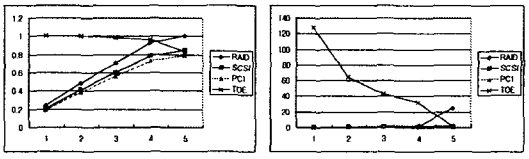
[그림 5]에서 보듯이, TOE를 추가시킴에 따라, 활용도와 대기열의 길이가 급속히 낮아지며 TOE를 5개로 한 경우에는 활용도가 60% 미만, 대기열의 길이는 0.78 까지 떨어져 안정적인 값을 찾아간다. 그러나 RAID, SCSI 및 PCI 버스의 활용도가 증가하게 된다. 즉, TOE 병목은 제거되었으나 다른 구성요소로 병목이 천이되었다. 따라서 천이된 병목을 제거하기 위하여, 최초 시스템 사양에 다음의 [표 6]과 같은 변화를 준다.

[표 19] 각 구성요소들의 조건 표

구성요소	조건
PCI 버스 수	1~2
PCI 버스 속도	66Mhz~133Mhz
RAID 수	2~4
SCSI 채널 수	1~2
SCSI 전송 속도	160MBps~320MBps
TOE 수	1~
TOE 파일 크기	1KBytes~64KBytes

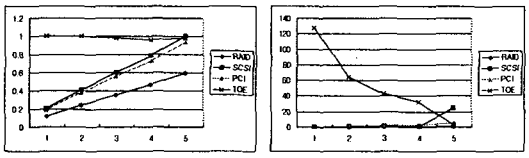
나. 병목제거를 위한 구성요소의 변화

[그림 6]은 SCSI 채널을 2개로 증가시킨 결과이다. SCSI의 활용도는 평균 0.846에서 0.568으로 33%정도 감소했으나, 대기열의 길이는 30.37에서 0.61로 98%나 감소하였다. 이는 Little의 결과(Little's Result)의 실제 예이다[13].



(a) 평균 활용도 변화 그래프 (b) 평균 대기열 변화 그래프

[그림 6] 각 구성요소의 평균 활용도 및 평균 대기열 그래프

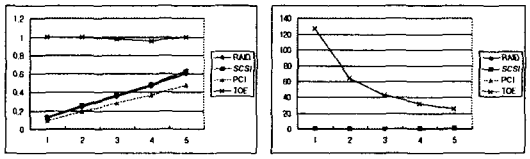


(a) 평균 활용도 변화 그래프 (b) 평균 대기열 변화 그래프

[그림 7] 각 구성요소의 평균 활용도 및 평균 대기열 그래프

SCSI의 성능 개선에 따라, 병목이 RAID로 이동하였으므로 그 개수를 증가시켜 병목을 제거한다. RAID의 수를 4개로 증가시킨 결과 [그림 7]과 같이 TOE의 활용도 및 대기열의 길이가 다시 증가하여 불안정해지기 시작했다. 그러나 앞에서 동일한 문제점에 대한 처리 방안을 구했으므로 더 이상 TOE의 개수를 늘리는 방안은 택하지 않는다.

대신, 병목인 SCSI의 속도를 우선 증가시킨 후, PCI가 새로운 병목으로 예상되므로, PCI의 성능을 개선하기로 한다. 그 후 마지막으로 TOE 파일 처리 크기를 64KBytes로 향상시키도록 한다.

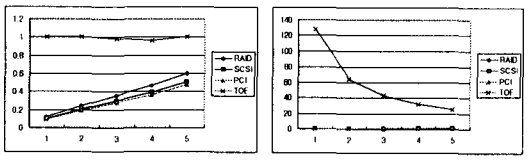


(a) 평균 활용도 변화 그래프 (b) 평균 대기열 변화 그래프

[그림 8] 각 구성요소의 평균 활용도 및 평균 대기열 그래프

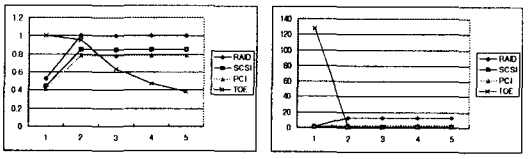
[그림 8]을 통해 SCSI의 안정성을 완전히 회복했으며, 병목이 TOE와 PCI 버스로 천이되었음을 알 수 있다. [그림 9]와 [그림 10]은 각각 PCI 수를 2개로 늘린 결과와 PCI 속도를 높인 결과이다. 외부적으로는 두 배의 성능 개선을 동일하게 적용하였지

만, 내부적으로는 66Mhz의 PCI 버스 2개, 133Mhz의 속도 PCI 버스 1개라는 상이한 성능 개선 방안을 적용한 결과이다. 따라서 동일한 의미의 성능 향상에서 서로 미묘한 차이만을 보일 뿐, 거의 흡사한 성능 향상을 나타내는 것이다.



(a) 평균 활용도 변화 그래프 (b) 평균 대기열 변화 그래프

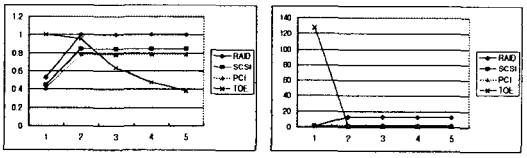
[그림 9] 각 구성요소의 평균 활용도 및 평균 대기열 그래프



(a) 평균 활용도 변화 그래프 (b) 평균 대기열 변화 그래프

[그림 10] 각 구성요소의 평균 활용도 및 평균 대기열 그래프

여기서 중요한 점은 PCI의 성능이 개선됨으로써 주변의 다른 장치들의 활용도와 대기열의 길이가 상승하여야 하는데, 실제로는 거의 변함이 없으며 오히려 [그림 10]에서는 SCSI의 활용도가 미세하게나마 낮아졌다. 이는 TOE를 제외한 모든 요소가 안정되었기 때문이다. 또한 이 결과는 잘 알려진 Amdahl의 법칙의 실제 예이다.[14] 즉, 장치의 수를 늘리는 것보다 그 장치의 성능을 배가 시키는 것이 더욱 효과적임을 알 수 있다.



(a) 평균 활용도 변화 그래프 (b) 평균 대기열 변화 그래프

[그림 11] 각 구성요소의 평균 활용도 및 평균 대기열 그래프

마지막으로 [그림 11]은 다른 구성요소들과 PCI 버스의 속도를 133Mhz로 고정시키고 TOE의 파일처리 크기를 64KBytes로 향상시킨 결과이다. 본 연구가 결국 다시 처음의 결과로 돌아왔음을 알 수 있다. 이것은 결국 병목이란 장치에 제한적인 조건을 명시하지 않으면 끝없이 반복적으로 천이되면서 나타나는 현상이라는 것을 나타낸다. 따라서 본 연구의 실험에서 무한 반복적인 병목현상 제거를 위한 성능 개선을 피하기 위해서는 NS의 요구 조건인 1Gbps 출력 속도에 맞추어 각 구성요소들의 평균 활용도 및 대기열 길이의 변화를 측정하여 분석해야 할 것이다. 결론적으로, RAID, SCSI 및 TOE의 활용도가 모두 감소하는 결과가 관찰되었다. 이는 시스템이 안정화되어 있다는 증거이다.

다. 시뮬레이션 결과 분석 결론

NS의 가장 기본적인 사양으로부터 시작하여 각 구성요소들의 성능과 그들이 주변의 요소들의 성능에 미치는 영향, 나아가 전체적인 성능에 미치는 영향을 분석하는 과정에서 병목은 시스템의 사양을 변화시킴에 따라 구성요소 간을 이동함을 볼 수 있었다. 그리고 그 병목을 완전히 제거한 NS의 구조를 얻을 수 있었다. 이제 NS의 전체적인 성능을 알아봄으로써 스트리밍 서버의 설계 목표를 달성할 수 있는 지 확인하는 절차만이 남아있다.

[표 7]에는 TOE의 출력 속도가 정리되어 있다. RAID를 4개, SCSI 채널을 2개, 버스를 2개, PMEM을 512MBytes로 한 경우 TOE는 수에 따라 50 ~ 241 MBps 또는 400~1,928 Mbps의 데이터가 하나의 NS로부터 출력된다. 즉 TOE를 3개 이상만 장착하면 스트리밍 서버의 설계 목표인 NS 1개당 1 Gbps의 전송속도를 달성할 수 있다.

[표 7] TOE의 출력 속도

RAID의 수	1	2	3	4	5
출력속도	50	100	147	192	250

(단위 : MBps)

6. 결 론

본 논문에서는 스트리밍 서버의 구조를 서비스 특성을 고려하여 시뮬레이션을 통해 분석하고 검증하는데 필요한 시뮬레이터를 구축하였다. 또한, 시뮬레이션을 수행하여 목표 시스템의 성능 분석과 각 구성요소들의 사양이 전체적인 성능은 물론 주변의 다른 구성요소에 미치는 상호 연관성을 분석하여, 본 논문에서 요구하는 조건을 충족시키는 최적의 NS의 구조를 제시하였다.

참고문헌

[1] Chervenak, A., Patterson, D., Katz, R., "Storage systems for movies-on-demand video servers", in Proc. IEEE Symp. on Mass Storage Systems, pp. 246-256 Sept. 1995.

[2] Wong, P., Lee, Y., "Redundant array of inexpensive servers (RAIS) for on-demand multimedia services", in Proc. IEEE Int'l Conf. on Communications, pp. 787-792 vol.2, 1997.

[3] J.Y.B. Lee, P.C. Wong, "Performance analysis of a pull-based parallel video server", in Proc. IEEE Conf. on Parallel and Distributed Systems, pp. 1217-1231, 2000.

[4] J.Y.B. Lee, "Buffer management and dimensioning for a pull-based parallel video server", in Proc. IEEE Conf. on Circuits and Systems for Video Technology, April 2001, pp. 485-496, 2001.

[5] M. Burns, A. George, B. Wallace, "Modeling and Simulative Performance Analysis of SMP and Clustered Computer Architectures," in Int'l Journal of Modeling and Simulation , Vol. 74, No. 2, pp. 84-96, 2000.

[6] A. George, J. Markwell, R. Fogarty, M. Miars, "An Integrated Simulation Environment for Parallel and Distributed System Prototyping," Int'l Journal of Modeling and Simulation , Vol. 72, No. 5, pp. 283-294, 1999.

[7] Kadrovach, B.A., Read, B.C., III, Young, F.C.D., Concha, L.M., Jarusewic, P., Jr., Pedersen, K., Bawcom, D., "Hardware Simulation with Software Modeling for Enhanced Architecture Performance Analysis", in Proc. IEEE National on NAECON, pp.454-461, 1998.

[8] S.G. Ziavras, "Performance analysis for an important class of parallel-processing networks", Parallel Architectures, Algorithms, and Networks, 1996., in Proc. 2nd Int'l Symp. on Parallel Architectures, pp. 500-506, 1996.

[9] Xu Ya, C. Orji, Deng Yi, N. Rishe, "An architecture for operating system support of distributed multimedia systems", in Proc. International Workshop on Multi Media Database Management Systems, pp. 56-63, 1995.

[10] Mequite Software, Inc., http://www.mesquite.com

[11] L. Kleinrock, "Queueing Systems Vol. I: Theory", Wiley, New York, 1975.

[12] J.D. Little, "A proof of the queueing formula L= λW," Op. Research, pp. 383-387, 1961

[13] G. Amdahl, "Validity of the single-processor approach to achieving large-scale computational capabilities", in Proc. AFIPS Conference Vol. 30, pp. 483, 1967.

[14] David E. Culler, Jaswinder Pal Singh "Parallel Computer Architecture", Morgan Kaufmann publishers, 1986.