

입출력 형태에 따른 다중처리기 시스템의 성능 분석

문 원 식*

An Analysis of Multi-processor System Performance Depending on the Input/Output Types

Moon Wonsik

〈Abstract〉

This study proposes a performance model of a shared bus multi-processor system and analyzes the effect of input/output types on system performance and overload of shared resources. This system performance model reflects the memory reference time in relation to the effect of input/output types on shared resources and the input/output processing time in relation to the input/output processor, disk buffer, and device standby places. In addition, it demonstrates the contribution of input/output types to system performance for comprehensive analysis of system performance. As the concept of workload in the probability theory and the presented model are utilized, the result of operating and analyzing the model in various conditions of processor capability, cache miss ratio, page fault ratio, disk buffer hit ratio (input/output processor and controller), memory access time, and input/output block size. A simulation is conducted to verify the analysis result.

Key Words : Multi-Processor, System Performance, Input/Output Type, Probability Theory

I. 서론

기존의 공유 버스 다중처리기 시스템의 성능 연구는 프로세서의 처리능력, 캐시의 일관성 문제와 병행한 성능연구, 버스의 구조 및 운영에 따른 성능 연구가 주를 이루었다[1-4]. 기존 모델들은 시스템의 전반적 성능이 아닌 특정한 부 시스템의 성능 분석이었으며, 입출력 성능연구나 입출력 수행에 의한 영향을 고려한 시스템 성능 모델의 구축은 활발치

못하였다.

본 논문에서는 입출력 부 시스템의 시스템 성능에 대한 기여도 및 공유 자원에 대한 과부하 영향을 제시하는 시스템의 성능 모델을 설정한다. 제시된 시스템 성능 모델을 통해 1) 단일 공유버스의 구조 및 운영 특성을 고려한 입출력 수행에 의한 공유 자원에서의 과부하 영향, 2) 입출력 수행의 시스템 작업 수행 성능에의 기여, 3) 버스 및 입출력 구조에 대기 장소를 설치한 파이프라인 식 운영 등을 분석한다.

* 평택대학교 컴퓨터학과 교수

본 논문의 구성은 다음과 같다. 2장에서는 입출력을 고려한 다중처리 시스템의 성능 지표와 중앙 및 입출력 프로세서의 작업부하를 확률적 방법에 의해 설정한다. 3장에서는 공유버스의 구조적 특성에 따른 성능 모델과 입출력에 의한 과부하를 분석하고, 4장에서는 디스크 입출력 처리과정 및 각 장치의 성능을 모델링한다. 5장에서는 분석적 모델의 구현 결과를 제시하며, 이를 시뮬레이션을 통해 검증한 후, 6장에서 결론으로 마감한다.

II. 기존 연구들

단일 프로세서 시스템의 성능은 캐시를 이용한 명령어 수행시간과 메모리 참조 시간의 합인 명령어 처리 시간으로 모델링되곤 했는데, Akella는 이에 입출력 전달 및 주변 장치 대기 시간의 입출력 처리 시간을 처음으로 포함하여 확장하였다[5].

Yang은 패킷 스위칭 멀티버스 구조를 갖는 CPU/memory 시스템의 성능 분석을 제공한다. 또한 메모리 입력부와 출력부에 무한 크기 버퍼가 있다는 구조를 가정한다. 동기와 비동기 모두에 대한 점근적 큐잉 네트워크 모델 들을 개발하였다[4].

Chiang 은 공유 버스 다중처리 운영을 위한 MVA(Mean Value Analysis) 분석적 모델의 정확성에 대한 연구를 수행하였고 명령어를 수행하는데 몇 개의 메모리 참조가 발생하는지 알아냈다[2, 6].

III. 시스템 성능 모델 및 작업 부하

다중처리의 성능은 그 시스템의 운영체제와 구조의 2가지 측면을 고려하여 분석해야 하나, 이들을 하나의 모델로 표현하는 것은 매우 복잡하므로 다음

과 같이 2단계로 나누어 모델링할 수 있다. 운영체제에 의한 특성은 상위 모델로, 구조적 특성에 따른 성능은 하위모델로 표현하고, 이 두 단계의 모델들은 서로 작업부하 및 성능인수를 공유 및 교환하여 하나의 시스템을 표현하도록 한다. 본 논문에서는 구조적 특성에 따른 하위 모델을 제시하며, 이는 추후 상위의 운영체제를 고려한 모델에 필요한 성능인수를 제공하게 된다.

3.1 시스템 성능 모델

본 논문에서는 다중처리 시스템의 구조적 특성을 고려하여 공유 자원에서 충돌에 의한 성능 영향 및 입출력 수행의 시스템 성능에 대한 영향과 각 부 시스템의 성능 역할에 대한 분석이 가능하도록 시스템 성능 모델을 확장한다. 이를 위하여 시스템 성능을 프로세서 대기시간, 입출력 요청 사이의 캐시를 이용한 프로세서 활동시간, 공유메모리의 참조시간 및 입출력 처리 시간의 합인 작업 동작주기(T_j)로 다음과 같이 정의한다.

$$\text{작업 동작주기}(T_j) = T_{jw} + (T_{pa} + T_{mr}) \times R_{mc} + T_{io}$$

T_{jw} : 프로세서 대기시간

T_{pa} (활동시간) : 지역 메모리를 이용한 프로세서 수행시간

T_{mr} (메모리 참조시간) : 메모리 요청의 처리 시간

R_{mc} : 입출력 요청사이의 중앙프로세서 공유메모리 참조 수

T_{io} (입출력 처리시간) : 입출력 요청의 처리 시간

위 모델에서 T_{jw} 는 상위 모델에서 다루므로 본 논문에서는 다루지 않는다. 이 기본 모델에 부 시스템

템들의 성능 특성을 나타낼 수 있도록 활동시간에는 프로세서의 처리 능력을, 메모리 참조시간에는 버스 및 메모리 모듈의 구조적 특성을, 입출력 처리 시간에는 입출력 프로세서, 디스크 버퍼 및 입출력 구조를 고려하여 아래와 같이 확장한다.

$$T_{pa} = T_{cr} \times R_c$$

T_{cr} : 캐시 참조 주기

R_c : 캐시 참조 수/활동시간

$$T_{mr} = T_{rw} + T_{ra} + T_{mw} + T_{ma} + T_{tw} + T_{ta}$$

T_{rw} : 요청 중재 시간

T_{ra} : 요청 전달 시간

T_{mw} : 메모리 대기 시간

T_{ma} : 메모리 접근 시간

T_{tw} : 자료 전달 중재 시간

T_{ta} : 자료 전달 시간

$$T_{io} = T_{iok} + T_{iop} + T_{ioc} + T_{iod}$$

T_{iok} : 커널 준비 시간

T_{iop} : 입출력 프로세서 처리 시간

T_{ioc} : 제어기 처리 시간

T_{iod} : 디스크 처리 시간

위의 각 성능 분석을 대입하면, 작업 동작주기는 아래와 같이 정의된다.

$$T_j = (T_{cr} \times R_c + (T_{rw} + T_{ra} + T_{mw} + T_{ma} + T_{tw} + T_{ta})) \times R_{mc} + (T_{iok} + T_{iop} + T_{ioc} + T_{iod})$$

3.2 중앙 및 입출력 프로세서의 작업부하 모델

중앙프로세서의 작업부하는 캐시 참조를 수행하는 활동 시간의 분포로 모델링할 수 있다. 한 명령어의 수행을 위해 평균 N_r 회 캐시 참조를 하며, 프로세서의 명령어 수행 능력이 K 명령어 수행 수/단위 시간 이라면, 평균 캐시 참조주기(T_{cr})는 $1/(N_r \times K)$ 이다. 프로세서의 활동 시간동안에 발생하는 캐시

참조 수의 합(R_c)은 매 캐시 참조마다 성공할 확률 P_c (캐시 미스 확률)로 공유 메모리를 참조하는 기하 (geometric) 분산으로 모델링되어, $R_c = (1 - P_c)/P_c$ 이 된다.

이로부터, 평균 프로세서 활동시간(T_{pa})은

$$T_{pa} = T_{cr} \times R_c = 1/(N_r \times K) \times (1 - P_c)/P_c$$

이 된다.

입출력 프로세서의 작업부하는 중앙 프로세서로부터의 디스크 입출력 요청 분포로 모델링한다. 즉, 각 중앙 프로세서들은 매 공유 메모리 참조마다 일정한 확률(P_m)로 입출력 요청을 발생한다고 모델링하면, 입출력 요청사이의 각 중앙 프로세서들의 메모리 요청 참조 합(R_{mc})은 $(1 - P_m)/P_m$ 이 된다.

실제 프로세서의 활동시간과 메모리 참조시간의 합($T_{pa} + T_{mr}$)의 주기마다 메모리 요청이 발생되므로, 각 중앙프로세서의 입출력 요구 발생 주기는 평균이 $(T_{pa} + T_{mr}) \times R_{mc}$ 이고 메모리리스(memoryless)한 분포를 갖는다.

기존의 공유메모리 및 버스의 요청률 및 충돌을 계산 시 입출력 수행의 영향을 무시하여 왔다. 실제로는 입출력 수행에 의해 추가적인 메모리 요청(이에 따른 버스 요청)을 하게 된다. 1회의 입출력 수행에 의한 메모리 요청 (R_{mio})은 $S_p/S_{sb} + R_{io}$ 로 표현되는데, 여기서 S_p 는 페이지의 크기, S_{sb} 는 시스템 버스의 크기, R_{io} 는 입출력 처리용 메모리 요청 수이다. 실제 N 개의 중앙프로세서 및 1개의 입출력 프로세서로 구성된 경우에 입출력프로세서의 작업 부하를 무시한 경우와, 중앙프로세서와 같다고 가정한 경우 및 앞에서 제시한 입출력 작업부하를 별도로 고려한 경우의 입출력 요청 주기 당 메모리 요청 수를 <표 1>에 보였다.

<표 1> 입출력 작업부하의 메모리 요청에 대한 영향

메모리 요청 수 \ 작업부하	N 중앙프로세서	$N + 1$ 중앙프로세서	N 중앙프로세서 1 입출력프로세서
중앙프로세서의 요청 수	$N \times (1 - P_m) / P_m$	$N \times (1 - P_m) / P_m$	$N \times (1 - P_m) / P_m$
입출력프로세서의 요청 수	0	$(1 - P_m) / P_m$	$N \times (S_p / S_{sb} + R_{io})$
총 요청 수	$N \times (1 - P_m) / P_m$	$(N + 1) \times (1 - P_m) / P_m$	$N \times (1 - P_m) / P_m + (S_p / S_{sb} + R_{io})$

IV. 공유 버스 특성에 따른 메모리 참조 모델

공유 버스의 동작 주기는 정상 읽기 요청에 대해 다음의 3단계로 나누어진다. 1) 요청전달 2) 메모리 접근 3) 자료전달. 여기서, 중재시간은 전달시간에 포함된 것으로 가정한다.

중형 컴퓨터에 적절한 단일 버스의 경우에도 스위칭(switching) 방식, 부 버스 구조 및 재요청을 위한 대기 장소 등의 개선을 통해 성능의 혁신적 향상을 이룰 수 있으므로, 이들을 고려하여 메모리 참조 모델을 구현한다.

패킷 스위칭(packet-switching)의 경우에는 실제 전달에 필요한 주기 동안만 버스가 점유되므로 (요청 전달시간 + 자료 전달시간)마다 최대 1개의 요청이 처리되어, 메모리 접근 시간에 비해 고속인 버스에 효율적이다. 이에, 독립적 부 버스 기능이 첨가되면 요청 전달은 주소 버스를, 자료 전달은 데이터 버스를 이용하며, 이들의 중재 및 전달이 서로 독립적이라면 부 버스 점유 주기마다 최대 1개의 요청이

처리된다. 메모리 참조가 파이프라인 식으로 이루어지기 위해서는 각 단계별로 공유 자원의 획득에 실패한 요청을 위한 대기 장소가 필요하다. 버스를 통과한 메모리 요청을 위한 메모리 모듈내의 입력 대기 장소(memory-in-buffer)와 메모리와 버스사이에 재요청 처리를 위한 출력 대기 장소(memory-out-buffer)의 활용으로 재요청의 처리가 효율적이 된다. <표 2>에 메모리 모듈내의 대기 장소에 따른 모델의 특성을 보인다.

본 논문에서는 앞에서 설명한 독립적 부 버스 및 메모리 모듈의 대기 큐를 고려한 비동기 패킷 스위칭 버스 운영의 메모리 참조모델을 Chiang의 사용자 MVA 모델 및[6], Yang의 동기/비동기 버스의 큐잉 모델[4]보다 간단한 큐잉 모델로 다음과 같이 제시한다.

단일 버스의 경우에는 동시에 버스를 통과한 요청의 수는 최대 1이므로, 각 메모리 모듈을 M/D/1 큐로 모델링할 수 있다. 실제로는 M/D/1 주소 버스를 통과한 출력이 마코비안(markovian)이 아니고,

<표 2> 대기 장소에 따른 메모리 모델의 특성

항목 \ 대기 장소	입력대기 장소	출력대기 장소	입력 및 출력대기 장소	대기 장소 없음
메모리점유시간	$T_{ma} + T_{tw}$	T_{ma}	T_{ma}	$T_{ma} + T_{tw}$
주소 버스 요청 수	메모리 요청 수	메모리 요청 수 + 재요청수	메모리 요청 수	메모리 요청 수 + 재요청수
메모리모듈의 모델	M/M/1	delay (constant)	M/D/1	delay (variable)

버스 주기 당 최대 1개의 요청만이 메모리 모듈로 전달되므로 이를 고려하여 수정하면,

$$T_{mw} = \frac{0.5 \times U_m}{1 - U_m} \times T_{ma} \times \frac{\left[1 - \left[\frac{T_{ra}}{T_{ma}}\right]\right]^2}{\sqrt{M}}$$

이 된다.

M : 메모리 모듈 수, U_m : 메모리 활용률

메모리 요청 주기의 동작이 항상 일정할 경우에는 T_{tw} 는 무시할 수 있으며, 그 이외의 요청 및 자료 전달이 독립적으로 이루어질 경우에는 $T_{tw} = T_{rw}$ 로 처리할 수 있다.

V. 디스크 입출력 성능 모델

다중 처리기에서 디스크 입출력 성능 분석은 2가지 측면에서 고려되어야 한다. 첫째는 작업 처리과정의 구성 성분으로 중앙프로세서의 입출력 요구로 인한 입출력 처리 과정을 분석해야 하고, 둘째는 입출력 수행의 운영체제 및 버스와 메모리 등의 공유 장치에 대한 과부하 영향(overhead)에 의한 시스템 성능에의 영향을 분석해야 한다. 이중 둘째 항은 앞에서 분석되었고, 이 장에서는 입출력 처리과정을 분석하도록 한다. 본 논문에서는 시스템 성능의 주요 병목요소인 입출력 성능의 향상을 위한 구조적 분석을 위하여, 입출력과정을 단계별로 분리 모델하고 입출력 성능에 가장 커다란 영향을 주는 디스크 버퍼의 성능 영향과 입출력 프로세서, 입출력 버스, 제어기, 디스크 등의 개별적 성능을 분석토록 한다. 입출력 프로세서가 시스템 및 입출력 버스로의 접근을 독립적으로 처리하도록 하고, 각 유닛에 비지에 대비한 명령 대기 장소를 가정한다. 이 경우에, 입출력 처리 과정은 각 유닛 별로 여러 단계로 나누어져 파이프라인 식으로 처리되는데, 실제 처리되는 과정

은 디스크 버퍼의 상태에 따라 다음의 3가지 경우로 나누고 각 경우의 발생 확률은 각각 P_1 , P_2 , P_3 이라하자.

첫 번째 경우는 입출력 프로세서의 디스크 버퍼에서 히트하여 입출력 제어기 또는 디스크의 직접 접근 없이 입출력 프로세서가 요청한 자료를 제공하는 경우이다.

두 번째 경우는 입출력 프로세서의 디스크 버퍼에서 미스(miss)하였으나 입출력 제어기의 디스크 버퍼에서 히트하여 실제 디스크의 접근이 필요 없는 경우, 세 번째 경우는 입출력 프로세서 및 입출력 제어기의 디스크 버퍼에서 모두 미스(miss)하여 실제 디스크 접근이 필요한 경우이다.

입출력 처리시간은 각 입출력 요청의 수행을 위한 입출력프로세서, 제어기 및 디스크에서의 대기시간과 수행시간의 합으로, 다음과 같이 구할 수 있다. 입출력 프로세서는 시스템 버스와 입출력 버스에서 동시에 접근이 가능하고, 각 장치는 대기 장소를 포함하고 있는 경우이다.

$$\text{입출력 처리시간}(T_{io}) = T_{iok} + T_{iop} + T_{ioc} + T_{iod}$$

T_{iok} (커널준비시간) : 입출력 요청의 전달 시간

T_{iop} (입출력프로세서처리시간) = 대기시간 + 점유시간

T_{ioc} (제어기 처리시간) = 대기시간 + 점유시간

T_{iod} (디스크 처리시간) = 대기시간 + 점유시간

위에 정의된 입출력 시간 성분들은 각 장치의 입출력 요청 당 점유시간, 활용도를 이용해 아래와 같이 구할 수 있는데, 입출력 버스의 대기 및 점유시간은 해당 명령 및 자료 전달 시간에 포함된다.

$$\text{장치 점유시간} = \frac{\sum(\text{점유시간}/\text{장치요구} \times P_i)}{\text{장치수}}$$

$$\text{장치 활용도} = \frac{\sum(\text{점유시간}/\text{장치요구} \times \text{입출력 요구율} \times P_i)}{\text{장치수}}$$

<표 3> 분석적 모델 및 시뮬레이션을 위한 파라미터 값들

환경 \ 파라미터	N	MIPS	P_c	P_m	P_1	P_2	$T_{ma}(usec)$	$S_{sb}(KB)$
(1)	4	40	0.1	0.001	0.1	0.1	0.16	2048
(2)	4	10	0.2	0.001	0.1	0.1	0.16	2048
(3)	4	10	0.1	0.0005	0.1	0.1	0.16	2048
(4)	4	10	0.1	0.001	0.5	0.1	0.16	2048
(5)	4	10	0.1	0.001	0.1	0.5	0.16	2048
(6)	4	10	0.1	0.001	0.1	0.1	0.16	1024
(7)	10	10	0.2	0.001	0.1	0.1	0.32	2048
(8)	10	40	0.1	0.0005	0.1	0.1	0.32	2048
(9)	10	40	0.1	0.001	0.1	0.1	0.16	1024
(10)	10	40	0.1	0.001	0.1	0.1	0.16	2048

여기서 \sum 는 입출력 처리 경우에 대한 합을 나타낸다. 각 장치의 활용도 및 점유시간으로부터, 각 장치의 대기시간은 입출력 프로세서와 제어기는 M/M/1 큐로, 디스크는 M/G/1로 모델링하여 각 큐의 대기시간으로 계산하고, 디스크의 재접속 시간은 재시도 횟수를 기하 분산을 이용하여 구할 수 있다.

VI. 모델의 수행 및 시뮬레이션

6.1 모델의 수행결과

본 논문의 실험에 사용된 시뮬레이터는 SMPL이라는 시뮬레이션 툴과 C로 작성한 것이다.

본 논문에서는 프로세서의 각 명령어들의 수행시간이 일정하고, 명령어의 캐시참조가 명령어 수행 중 특정시간에 발생한다고 가정하였다. 또한 공유메모리 및 디스크 입출력 요청은 각 장치에서의 읽기만을 고려하였고, 디스크를 제외한 다른 입출력은 고려하지 않았다.

본 논문에서 제시한 분석적 모델들을 프로세서처

리 능력, 캐시미스 율, 페이지 폴트 율, 디스크 버퍼 미스 율(입출력 프로세서), 디스크 버퍼 미스 율(제어기), 메모리 접근 시간, 입출력 블록 크기 등의 다양한 조건하에서 수행하였다. 마찬가지로 시뮬레이션을 위해서도 동일한 값으로 하였으며 이를 위한 파라미터 값 들은 논문 [2, 5, 7]에서 사용한 값들을 가지고 만들었으며 <표 3>에 제시하였다.

분석적 모델의 적합성을 확인하기 위하여, 동일한 가정 하에서 확률분포를 이용한 시뮬레이션 모델을 수행한 결과가 <표 4>이다. <표 4>에서 A는 분석적 모델의 결과이고 S는 시뮬레이션 모델의 결과이다. 시뮬레이션 결과와 분석적 모델의 결과가 매우 비슷한 결과를 보여주어 본 논문에서 제안한 분석적 모델이 매우 정교하다는 것을 보여 주고 있다.

6.2 분석결과

본 논문에서 제시한 파라미터 값과 성능 모델 분석을 통해 다음과 같은 성능적 특성을 확인할 수 있다.

① 입출력 수행의 공유자원에 대한 영향이 버스 및 메모리 참조시간 모델에 고려되어야 한다. 실제

<표 4> 분석적 모델 및 시뮬레이션 수행 결과

환경		Time				Utilization		R_{mio}
		$T_{pa} \times R_{mc}$	$T_{mr} \times R_{mc}$	T_{io}	T_j	U_{iop}	U_{iod}	R_m
(1)	A	1.88	0.34	0.33	2.55	33.7	11.4	0.20
	S	1.87	0.34	0.32	2.53	33.8	11.5	0.21
(2)	A	3.33	0.33	0.29	3.95	20.2	6.9	0.20
	S	3.22	0.32	0.29	3.83	20.2	6.9	0.21
(3)	A	14.96	0.64	0.26	15.86	4.7	1.6	0.12
	S	14.86	2.07	0.26	15.79	4.7	1.6	0.12
(4)	A	7.50	0.32	0.64	8.46	9.4	16.3	0.20
	S	7.47	0.32	0.61	8.40	9.4	16.4	0.20
(5)	A	7.50	0.32	0.60	8.42	9.4	16.2	0.20
	S	7.47	0.32	0.59	8.38	9.4	16.3	0.20
(6)	A	7.48	0.32	0.22	8.02	7.24	3.2	0.20
	S	7.44	0.32	0.22	7.98	7.24	3.2	0.21
(7)	A	3.33	0.60	0.62	4.55	64.8	16.2	0.20
	S	3.46	0.65	0.59	4.70	65.4	16.3	0.20
(8)	A	3.77	1.45	0.59	5.81	55.3	12.1	0.11
	S	3.76	1.52	0.56	5.84	55.9	12.0	0.11
(9)	A	1.79	0.38	0.42	2.69	65.8	28.1	0.20
	S	1.85	0.37	0.39	2.61	65.9	27.9	0.21
(10)	A	1.87	0.39	1.18	3.44	88.0	28.7	0.20
	S	1.82	0.40	1.06	3.28	87.9	28.1	0.20

메모리 요청에 대한 입출력 영향은 11~21%를 차지한다.

② 시스템 성능에 대한 입출력 구성비를 고려해야 한다. 입출력 요청 중 실제 디스크 접근이 1%인 경우에도 입출력 처리시간이 최대 34%를 차지한다.

③ 시스템 성능에 대한 가장 큰 구조적 영향은 효율적인 버퍼링이다. 즉, 캐시 미스 율, 페이지 폴트 율, 및 디스크 버퍼의 미스 율을 줄이는 것이 가장 확실한 성능 향상 방법이다.

④ 공유 자원의 대기 장소에 의해 처리과정이 파이프라인 식으로 되며, 이외의 모델도 큐잉 이론을 이용해 쉽게 된다.

⑤ 버스의 특성(점유방식, 독립적 부 버스 등)이

버스의 처리율에 큰 영향을 준다.

VII. 결론

본 논문에서는 공유버스 다중처리기의 작업부하를 확률적으로 설정한 분석적 모델을 제안하였다. 이 모델은 Ching의 사용자 MVA 모델 및 Yang의 동기/비동기 버스의 큐잉 모델보다 간단하다. 분석적 모델이 실제 시스템에 잘 적용될 수 있고 잘 만들어졌다는 것을 보이기 위하여 시뮬레이션을 수행한 결과와 비교하였는데 거의 비슷한 결과를 얻을 수 있다는 것을 보였다.

시스템 성능 모델에는 단일 버스의 성능향상을 위한 패킷 스위칭 버스, 독립적 부 버스 및 메모리 대기 큐의 3가지 구조적 특성과 입출력의 영향을 고려한 메모리 참조 모델과 입출력프로세서 및 디스크 버퍼를 고려한 입출력 처리 모델 및 이들의 시스템 성능에 대한 영향을 포함하였다. 이로부터, 시스템의 작업 수행 과정 및 입출력 수행의 시스템 성능에 대한 기여도 등에 관한 포괄적 분석이 가능하다. 버스, 메모리, 입출력 장치 등의 요청된 자원이 비지거나 동시에 다중요청이 발생할 경우, 새로운 요청의 발생 없이 대기할 수 있는 대기 장소를 갖는 구조의 성능적 특성을 큐잉모델로 분석하였다. 프로세서의 처리 능력, 입출력 처리 능력 및 버스의 구조적 개선에 따른 시스템의 성능을 캐시, 공유 메모리 및 디스크 버퍼의 히트 율에 따라 모델을 수행하고 그 결과를 분석하였다.

본 모델에서는 디스크 캐시가 계층적으로 구성되고 그의 히트 율이 매우 높은 경우에 대해 분석했는데, 실제 매우 큰 파일 시스템을 관리하는 경우에는 이의 히트 율이 극히 저조하므로 다양한 작업 부하의 특성 연구가 필요하다. 본 연구에서 얻은 성능 성분들을 활용하여 다중처리기 운영체제의 특성을 고려한 시스템의 상위 성능 모델의 연구로 확장 수행되어야 할 것이다.

참고문헌

- [1] Wen-Tsuen Chen and Jang-Ping Sheu, "Performance Analysis of Multiple Bus Interconnection Networks with Hierarchical Requesting Model," IEEE Trans. on Computer, vol. 50, No. 7, pp. 834-842.
- [2] Men-Chow Chiang and Gurindar S. Sohi, "Experience with Mean Value Analysis Models for Evaluating Shared Bus, Throughput-Oriented Multiprocessors," Performance Evaluating Review, vol. 19, May 2003, pp. 90-102.
- [3] J. L. Grino et al., "Analysis and Simulation of Multiplexed Single-Bus Networks with and without Buffering," Proc. of Symposium on Computer Architecture, 1995, pp. 414-421.
- [4] Qing Yang and Laxmi N. Bhuyan, "Analysis of Packet-Switched Multiple-Bus Multiprocessor Systems," IEEE Trans. on Computer, vol. 50, March 2001, pp.352-357.
- [5] Janaki Akella and Daniel P. Siewiorek, "Modeling and Measurement of the Impact of Input and Output on System Performance," Proc. of Symposium on Computer Architecture, 2005, pp. 390-399.
- [6] Men-Chow Chiang and Gurindar S. Sohi, "Evaluating Design Choices for Shared Bus Multiprocessors in a Throughput-Oriented Environment," IEEE Trans. on Computer, vol. 51, No. 3, pp. 297-317.
- [7] Per Stenstrom, "A Survey of Cache Coherence Schemes for Multiprocessors," IEEE Computer, June 2000, pp. 12-24.
- [8] 장범환, "트래픽 세션의 포트 역할을 이용한 네트워크 공격 시각화," 디지털산업정보학회 논문지, 11권 4호, 2015, pp.47-60.
- [9] 양환석, "Mobile Ad-hoc Network에서 영역 기반 보안 멀티캐스트 기법 연구," 디지털산업정보학회 논문지, 12권 3호, 2016, pp.75-85.

■ 저자소개 ■



문 원 식
Moon Wonsik

1995년 3월 ~ 현재
평택대학교 컴퓨터학과 교수
1995년 2월 동국대학교 컴퓨터공학과
(공학박사)
1989년 2월 동국대학교 컴퓨터공학과
(공학석사)
1987년 2월 동국대학교 수학과(이학사)

관심분야 : 병렬처리, 분산알고리즘
E-mail : moonws@ptu.ac.kr

논문접수일 : 2016년 11월 21일
수 정 일 : 2016년 12월 8일
게재확정일 : 2016년 12월 13일