論 文

확률적 컴퓨터 성능평가 모델설정에 관한 연구

正會員 金 相 福* 正會員 金 正 祺**

A Study on the Construction of the Stochastic Model for the Computer Systems Performance Evaluation

Sang Bok KIM*, Jung Ki KIM** *Regular Members*

要 約 본 논문은 benchmark program의 명령어 mix들과 이 명령어mix 들의 분포와 빈도들을 parameter로 해서 컴퓨터 성능평가를 행할 수 있는 확률적 모델을 제시한다. 이 모델을 Intel 8086/8083마이크로 프로세서의 성능 평가에 적용시켜 봄으로써, 현존하는 컴퓨터시스템뿐만 아니라 미개발 시스템을 위한 성능 평가 모델로도 가능하다는 것을 보였다.

ABSTRACT This paper constructs a stochastic model for computer performance evaluation which has several parameters such as the kinds of instruction mix of benchmark programs, distribution and frequency of instruction mix.

It shows, by applying the model to the performance evaluation of the Intel 8086/8088 microprocessor, that this model could be utilited not only for performance evaluation of existing computer systems but also for estimation of nonexisting systems.

1. 서 론

컴퓨터 성능평가는 첫째는 특정 컴퓨터 시스템을 구입하는 것이 가격과 성능면에서 가장 적합한가를 판단하는데 상용되며 둘째로는 개발하고자 하는 컴퓨터 시스템의 성능을 추정하는데 사용된다.

널리 사용되고 있는 성능평가 방법으로써 명령어 Mix 방법과 벤치마크 프로그램에 의한 방법들이 있는데, Mix 방법들은[1],[2],[3] 해당 프로그램 상에서의 명령어 mix의 실행횟수의 빈도수를 성능평가에 이용한다. 이 방법은 1950 년 후반 Gibson에 의한 연구[3]로부터 시작되었다. Gibson 분류에 따라 임의의 명령어 세트를 분류하고 그 각각의 분류에 대한 평균 실행비에

*慶尙大學校 電算統計學科
Gyeongsang University, Dept. of Computer Sciences
**中央大學校 電子工學科
Dept. of Electronic Engineering Chungang University Seoul, 151 Korea.
論文番號 : 89-06 (1988. 10. 26)

의거하여 산출된 명령어 실행비는 서로 다른 컴퓨터들의 성능을 비교하는데 사용할 수 있다. 그러나 이 방법의 문제점은 많은 명령어들을 분류하기가 힘들며, 설사 분류가 가능하더라도 주소변경에 기인되는 다양한 실행시간, 다양한 피연산자 및 명령어 실행에 수반되는 데이타들의 종속성으로 인하여 컴퓨터의 수행성능을 평가하는 것이 어렵다. 이런 방법의 적용은 일반적 관점에서의 성능평가에는 기여하는 바가 적지만 사용한 알고리즘들의 복잡한 분석이나 같은 계열의 컴퓨터 간의 성능비교에서 이들이 유용해질 수도 있다.

벤치마크 프로그램에 의한 성능평가 방법이란 Whetstone[4], Dhystone[5], Unix 벤치마크[6] 등과 같은 공인된 프로그램들을 비교하고자 하는 시스템에 수행시켜 서로의 성능을 비교하는 방법인데 이같은 벤치마크 시험 방법을 이용하여 시스템을 선정하고자 할 때는 주의를 기울여야 한다. 왜냐하면 각 컴퓨터 제조회사의 시스템에게 유리한 특정 벤치마크 프로그램의 결과를 성능평가 자료로 이용하기 때문이다. 그 단적인 예로써, Bell et al은 VAX 11/780과 DEC 시스템 2060상에서 여러개의 벤치마크 프로그램들을 수행시켜 본 결과[7]를 보면, 하나의 벤치마크 프로그램 상에서는 VAX 11/780이 약 3배 느리지만, 다른 벤치마크 프로그램 상에서는 약 50%나 빠르다는 사실을 발견했다. 더구나 특정 컴퓨터에 대하여 서로 다른 벤치마크 프로그램들을 수행시키면 상이한 결과를 초래할수도 있다. 예로써 VAX 8600과 VAX 785상에서 McCallum's simple test를 수행시켜 본 결과 Whetstone과 Sieve 벤치마크로 부터 얻은 결과와 35%의 차이[8]를 보였다.

명령어 mix 방법과 벤치마크 프로그램 방법은 근본적인 차이점[9]이 있다. 전자는 오직 CPU의 성능을 조사하며 후자는 컴파일러와 CPU의 성능을 조합해서 조사한다. 컴파일러는 CPU성능을 측정하는 것보다도 많은 경우에서 성능평가 결과의 큰 변화를 야기시킨다[10].

이들은 서로 다른 시스템들의 성능비교 목적을 위해 CPU의 성능평가에 유효하게 사용되긴 하지만 이들 두방법들을 하나로 조합해서 사용할 시 야기되는 성능평가의 신뢰성이 감소하게 된다.

따라서 본 논문에서는 위에서 언급한 두가지 성능평가 방법들을 통합하여 하나의 성능평가 모델을 설정한 후 그 유효성을 입증하고자 한다. 이 모델은 통계적인 방법인 마르코프 체인을 이용하여 설정되었으며 이 모델의 매개변수들로써는 벤치마크 프로그램 상에서의 명령어 분류를 명령어 Mix로 사용하여 이들 각 명령어들의 사용 분포 빈도와 각각의 명령어들이 수행하는데 소요되는 클럭펄스 수를 사용하였다. 이 모델의 유효성을 입증하고자 벤치마크 프로그램으로 널리 사용되고 있는 prime number를 찾는 프로그램과 finonicci 수열을 구하는 프로그램을 Intel 8086/8088 마이크로 프로세서를 위한 어셈블리 언어로 작성하여 그 유효성을 입증하였다.

Ⅱ. 성능평가 모델

이 모델은 연속적인 명령어 수행 상태 묘사에 적합한 마르코프체인[11], [12], [13]을 이용하여 해석한 후 Intel 8086/8088 마이크로 프로세서 상에서 수행 가능한 어셈블리 언어로 작성된 프로그램을 대상으로 이 모델의 유효성을 입증하였다.

이 모델을 구성하는데 필요한 상태 추이도는 각 프로그램에서 수행되어지는 명령어들의 수행상태를 의미하며, 이들 상태의 수는 각 명령어 Mix들이 수행하는데 필요한 클럭펄스 수로써 구성하였고 상태 추이 확률은 시스템 상에서 수행되는 프로그램들의 명령어 Mix들의 분포 빈도들을 사용하였다.

Ⅱ - 1. 초기 모델

초기 모델은 프로그램에서의 명령어 Mix들의 분포 확률을 구할 수 있다는 가정하에 그림 1과 같이 구성했다. 그림 1에서의 상태 2, 상태 3, …, 상태n, 상태B 각각은 명령어 Mix들의 수

행 상태를 의미하며 상태 2은 2개의 클럭 펄스를 요하고 상태 3은 3개의 클럭 펄스의 수를 상태n는 n개의 클럭펄스를 요하는 명령어 Mix들의 상태를 나타내는 비분기 명령어들의 수행 상태를 상태B는 분기 명령어의 수행 상태를 의미한다.

이들 프로그램이 시작 후 해당되는 명령어 M Mix를 수행할 초기확률은 P_{s2}, P_{s3}, ……, P_{sN}, P_{sB} (S는 프로그램이 처음 시작될 때를 의미한다.)
로써 이 확률은 프로그램 상에서의 명령어 Mix 분포비로 주어진다.

그림 1에서의 상태 S/E는 프로그램이 시작되어지는 시작점을 나타냄과 동시에 프로그램이

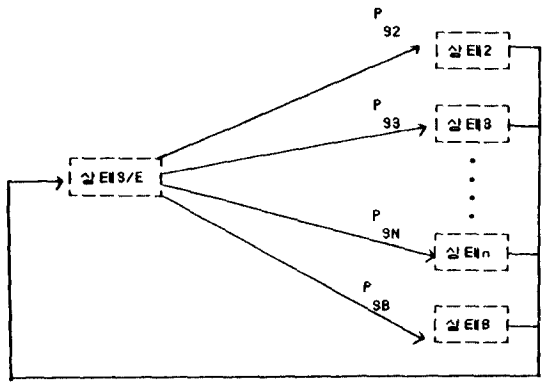


그림 1 Initial model

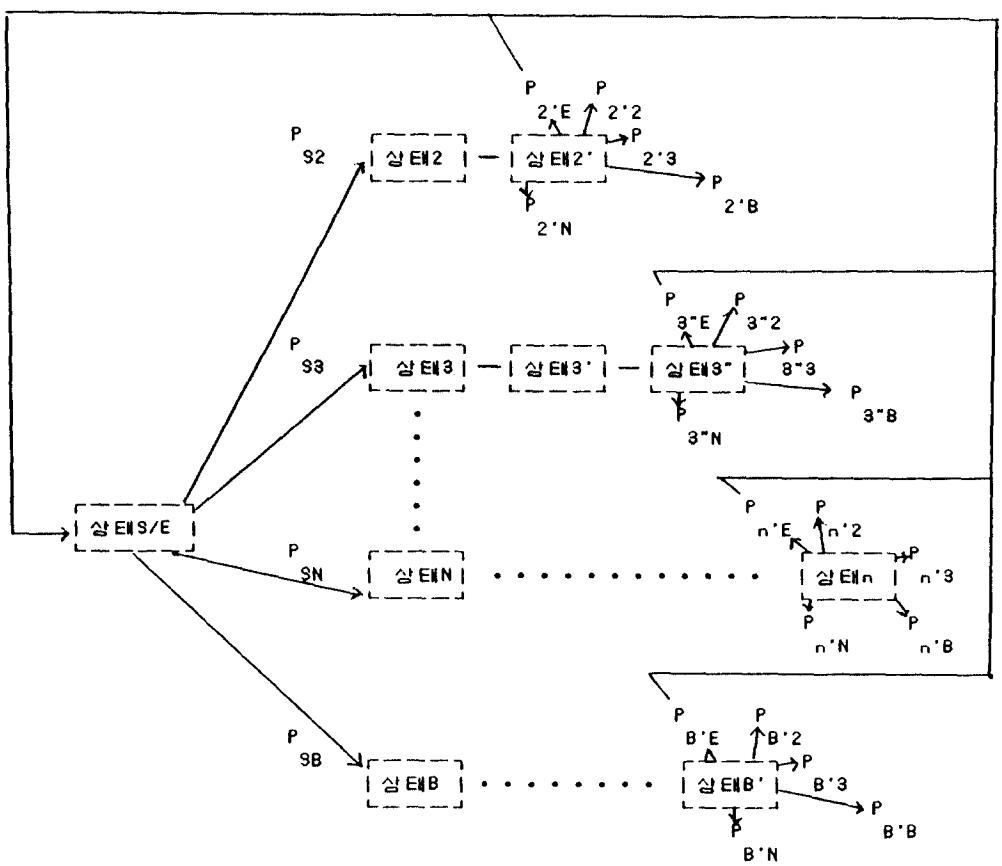


그림 2 Modified model

수행을 완료하고난 뒤에 END 문을 만나면 다시 상태S/E로 돌아온다는 것을 나타내기 위해 상태 S/E로써 표기했고, 상태2, …, 상태n 에 해당되는 명령어들을 수행상태수는 Intel 8086/8088 마이크로 프로세서 상에서 수행에 필요한 클럭펄스 수만큼의 상태수를 나타내었으며, 각 상태들의 상태수는 해당되는 명령어 수행에 필요한 클럭펄스 수에 대응시킨다면 그림1은 그림2와 같이 확장할 수 있다.

그림2에서 상태2, 상태3, …, 상태n, 상태B는 상태 S/E에서 수행 가능한 명령어들의 Mix들을 나타내었으며, 이들을 만난 다음 각 명령어들은 수행하는데 필요한 클럭펄스 수에 해당되는 상태들로 전이가 이루어진다. 예로써, Intel 8086/8088 마이크로 프로세서 상에서의 INC 명령어가 수행하는데 필요한 클럭 펄스의 수는 두개이므로 이 INC 명령어에 해당되는 상태는 둘로 나타낸다. 이들이 수행에 필요한 클럭펄스의 수만큼 전이를 한다음 즉 하나의 명령어 Mix를 수행한 후 다음 수행할 명령어 Mix를 만날 확률은 그림2의 확률들로써 이는 프로그램에서의 각 명령어 Mix들의 분포 빈도에 해당된다.

그러나 그림2에서 현재 수행한 명령어 Mix가 분기 명령어일 경우 이 명령어를 수행한 후 만날 다음 명령어 Mix는 달라져야한다. 왜냐하면 많은 프로그램들을 분석해 보면 분기 명령어 다음에 곧 바로 또 다른 분기 명령어를 만나는 경우는 극히 희박하므로 그림3과 같이 보다 간단한 성능평가 모델을 설정하는 것이 바람직하다.

Ⅱ-2. 개선된 모델

그림2에서 현재 수행한 명령어가 분기 명령어일 경우 곧 바로 또다른 분기 명령어를 만날 확률은 동적 프로그램 상에서 희박하므로 이 경우 다음 상태로의 전이 확률에서 P′ (여기서 B는 분기 명령어를 의미)는 제로(0)로 두었다.

따라서 그림2를 보다 더 간단한 모델로 구성하기 위하여 분기 명령어 다음에 반드시 비분기 명령어가 오도록 함으로써 그림3과 같은 개선된 모델로 만들 수 있으며 이 그림 3으로 부터 얻을 수 있는 마르코프 체인의 상태 추이확률은 표1과 같다.

그림3의 상태 추이도로부터 직접 구할 수 있는 양은 상태 S/E를 출발한 후 해당되는 상태들인 프로그램 상의 명령어들을 수행한 후 다시 상태 S/E로 되돌아올때까지 방문한 상태수 들이 바로 귀환시간 상되는데 본 논문에서는 이들 방문시간의 총계가 바로 프로그램이 수행되는데 필요한 클럭펄스의 수인 것이다.

표1로부터 평균 귀환 시간을 구하기 위해 관련되는 첫번째 양은 각 상태들에 존재하는 정상 상태 확률 벡타A인데, 이 벡타 A는 각행이,

$a = \{a1, a2, a3, \cdots\cdots, ai\}$ $(i = 1, 2, 3, \cdots, n)$를 가지며 이 상태 확률중의 어떤 벡타 a에 대해 한 단계후의 상태 확률들의 벡타는 ap이므로, 정상 상태 확률 벡타는 $ap = a$를 만족하는 벡타a이다. 그리고 그때 어떤 상태로 되돌아 오는 평균 귀환 시간은 아래에 있는 마르코프 체인의 fundamental matrix [11], [13]에 의해 구할 수 있다.

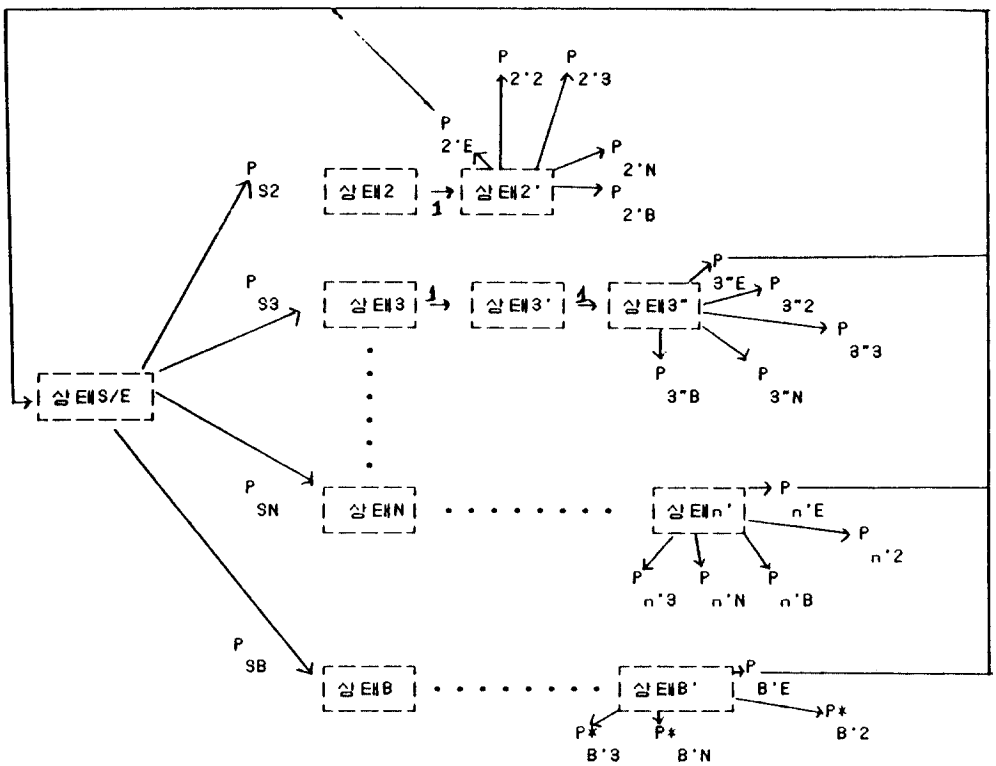
$$Z = (I - (P - A))^{-1} \quad (1)$$

여기서 A는 모든 행이 상태 확률 벡타a를 가진다.
I는 단위 행렬이다.

그때 평균 귀환 시간의 행렬은

$$M = (I - Z + E * Z1) * D \quad (2)$$

여기서 Z1은 식(1)에서의 fundamental matrix Z의 대각을 제외한 모든 요소들을 0으로 가지는 대각 행렬이다.
그리고 D는 $d(i, i) = 1/a(i)$의 요소들을 가지는 대각 행렬이다.



여기서 $P^*_{B'2}$, $P^*_{B'3}$, $P^*_{B'N}$의 확률들은 프로그램에서 비분기 명령어들의 분포 빈도에 의해 주어진다.

그림 3 State transition diagram

표 1 State transition probability matrix.

	상태 2	상태 3 – 상태N	상태B	상태S/E	상태2′	상태3′	상태3″	상태n′	상태B′
상태 2	0	0	0	0	0	1	0	0	0	0
상태 3	0	0	0	0	0	0	1	0	0	0
상태N	0	0	0	0	0	0	0	0	1	0
상태B	0	0	0	0	0	0	0	0	0	1
상태S/E	P_{S2}	P_{S3}	P_{SN}	P_{SB}	0	0	0	0	0	0
상태 2′	$P'_{2'2}$	$P'_{2'3}$	$P'_{2'N}$	$P'_{2'B}$	$P_{2'E}$	0	0	0	0	0
상태 3′	0	0	0	0	0	0	0	1	0	0
상태 3″	$P'_{3''2}$	$P'_{3''3}$	$P'_{3''N}$	$P'_{3''B}$	$P_{3''E}$	0	0	0	0	0
상태n′	$P'_{n'2}$	$P'_{n'3}$	$P'_{n'N}$	$P'_{n'B}$	$P_{n'E}$	0	0	0	0	0
상태B′	$P^*_{B'2}$	$P^*_{B'3}$	$P^*_{B'N}$	0	$P_{B'E}$	0	0	0	0	0

Ⅲ. 성능평가 모델의 유효성 입증

성능평가 모델의 유효성을 입증하기 위해 Intel 8086/8088 어셈블리 언어로 작성된 prime number를 구하는 프로그램과 5백만 이하의 fibonacci 수열을 구하는 프로그램을 추적 조사한 실제값과 모델에 의한 측정값을 표(2)에 나타내었다. 표에서 볼 수 있듯이 실제값과 모델에 의한 측정값이 거의 일치함을 볼 수 있다.

표 2 Mearsured value and true value

	측정값	실제값
prime number	227318 클럭펄스	218212 클럭펄스
finanicci 수열	879 클럭펄스	862 클럭펄스

Ⅳ. 결 론

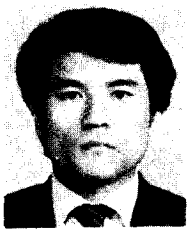
컴퓨터 성능을 평가하는 여러 방법들이 나름대로의 파라미터들을 이용하여 같은 계열이나 혹은 다른 기종들간의 성능을 평가하고 있으나, 이들의 공통된 단점은 복잡한 파라미터를 사용하고 있을 뿐만 아니라 이들 방법에 의한 결과의 신빙성이 약하다는 것이다.

본 논문에서는 시스템상에서 수행 가능한 대상 프로그램의 명령어 분포빈도와 수행되는 해당 명령어들의 클럭 펄스의 수만을 매개변수로 한 성능평가 모델을 제시하고 이를 통계적 방법인 마르코프 체인을 이용하여 모델을 설정한 후 두개의 프로그램 만을 대상으로 Intel 8086/8088 마이크로 프로세서 상에서 모델의 유효성을 입증하였다. 그러나 Intel8086/8088 마이크로 프로세서 시스템 뿐만 아니라 명령어 수행에 필요한 클럭펄스의 정확한 정보만 주어 진다면 모든 기종에 이 모델이 적용 가능하다고 사료된다.

본 논문에서 제시한 이 모델은 대상 프로그램의 수행의 성능평가를 클럭펄스의 총 수로 결과를 나타내게 하는 확률적인 방식으로 설정된 만큼 각종 컴퓨터 시스템에서 운용되는 명령어 수행의 확률적인 분포의 정확성이 모델에 의한 성능평가의 정확도를 결정할 수 있다.

參 考 文 獻

1. D.E. Knuth, "An Empirical study of Fortran Programs", ,Software P & E. 1971. pp. 105-133.
2. R.A. Arbuckle, "Computer Analysis and Thruput Evaluation", Computers and Automation, Jan., 1966, pp. 12-19.
3. J.C. Gibson, "The Gibson Mix", IBM tech. report TR2 043, Jun 18, 1970.
4. H.J. Curnow and B.A. Wichmann, "A Synthetic benchmark", The Computer Journal Feb. 1977, pp. 43-49.
5. R.P. Weicker, "Dhrystqns: A Synthetic Systems Programming Benchmaek", Comm. ACM. Oct. 1984 , pp. 1013-1030.
6. D.F. Hinnant, "Benchmarking UNIX Systems, "Byte, Aug., 1984, pp. 132-409.
7. C.G. Bell, J.C. Mudge, and J.E. McNamara, "Computer Engineering, ADEC view of Hardware Systems Design, Digital Press, Burlington, Mass., 1978.
8. J.C. McCallum, "benchmark Results for Micro computers and Large Computers", Data Processing, Oct. 1986, pp. 426-433.
9. B. Randell and L.J. Russell, "ALGOL 60 Implementation", Academic Press, Orlando, Florida, 1964.
10. J. Gilbreath and G. Gilbreath, "Eratosthenes Revisited", Byte, Jan. 1983. pp. 283-326.
11. J.G. Kemeny and R.O. Winder, "Finite Markov chains", Chapter 4, Van Nostrand Reinhold Company, New York, 1960.
12. M. Ajmone Marsan, G. Balbo, and G. Conte, "Performance Models of Multiprocessor Systems", The MT Press, 1986.
13. H. KOBAYASHI, "Modeling and Analysis", ADDISON-WESLEY PUBLISHING COMPANY, 1978.



金 相 福(Sang Bok KIM) 正會員

1955年 2月 3日生

1979年：中央大學校 電子工學 學士

1981年：中央大學校 電子工學 碩士

1989年：中央大學校 電子工學 博士

1984年 3月：慶尚大學校 電子計算 學科 ~現在 助教授

金 正 祺(Jung Ki KiM) 正會員

1942年 5月 5日生

1965年 2月：延世大學校電氣工學科卒業

1969年 2月：延世大學校大學院電氣工學科(工學碩士)

1975年 2月：延世大學校大學院電氣工學科(工學博士)

1970年 3月~1977年 2月：光云工科大學 副教授

1977年 3月~現在：中央大學校電子工學科教授

1982年 12月~1983年 12月：美亞里大學校客員教授