

다중프로세서 시스템을 위한 동적스케줄링방침의 최적화에 관한 연구

오종진*, 조용환*, 백두권**

* 충북대학교 전자계산기공학과, ** 고려대학교 전산과학과

Optimization on the Dynamic Scheduling Policy of Multiple Processor Systems

Jong Jin Oh*, Yong Hwan Cho*, Doo Kwon Baik**

* Dept. of Computer Engineering, Chungbuk National Univ.

** Dept. of Computer Science, Korea Univ.

ABSTRACT

This study has considered several kinds of system environments regarding to job classes and the number of processors, in order to schedule jobs to processors in a multiple processor system with dynamic scheduling policy. For each environment, many simulations have done and showed optimal time length on feedback interval, by which the system maximizes the number of jobs to be processed in system within a given time.

1. 서론

최근 하드웨어(Hardware)의 급속한 성장으로 소형의 마이크로프로세서(Microprocessor)들을 적은비용으로 이용할 수 있게 됨에 따라, 다수의 프로세서들을 결합시켜 보다 많은 작업들을 처리케 함으로써 시스템의 처리속도와 신뢰성을 증가시키는 다중프로세서시스템(Multiple Processor System)의 이용이 증대되고 있다. 이러한 다중프로세서시스템에 있어서 프로세서들에 대한 Job들의 효율적인 스케줄링(Scheduling)은 시스템의 성능에 커다란 영향을 미치고 있다.

동적스케줄링방침(Dynamic Scheduling Policy) [4], [6]은 프로세서들의 상태에 관한 Feedback Information을 주기적으로 이용함으로써 스케줄링의 중간과정에서 이루어지는 시스템의 상태에 따라 Job들을 프로세서에 동적으로 할당시키는 것이다. 이것은 미리 정해진 방법에 따라 Job들을 프로세서에 할당시키는 정적스케줄링방침(Static Scheduling Policy) [2], [3]과는 크게 대조적이다.

본 연구는 동적스케줄링 방침을 위해, Job Class 및 프로세서의 수에 따른 여러가지 시스템환경의 자각에 대하여 시뮬레이션(Simulation)을 반복적으로 수행시킨 결과를 분석함으로써, 각 시스템환경에 따른 Feedback 정보의 최적 제공시간간격을 구하고자 하는데 그 목적이 있다.

이를 위하여, 본 연구에서는 Informal Model Description을 이용하여 동적스케줄링방침의 적용을 위한 다중프로세서시스템의 모형을 정립하였으며, 이것을 기반으로한 네트워크 모델 어프로치에 의한 시뮬레이션을 수행하여 그 결과를 분석하였다.

2. 동적스케줄링방침의 개요

동적스케줄링 방침은 프로세서의 현재상태에 관한 Feedback Information의 적절한 제공으로 시스템에서 발생하는 요구에 따라 프로세서들에 대한 Job의 스케줄링을 변화시킬 수 있도록 해준다. 따라서 시스템환경에 따라 이 방침은 스케줄링의 중간과정에서 발생하는 임의의 상황에 능동적으로 Job들을 할당시킴으로써, 시스템의 효율을 극대화시킬 수 있다.

동적스케줄링 방침은 정적스케줄링방침에 반하여 Job의 도착이나 프로세서의 서비스능력에 관한 선점적지식(A Priori Knowledge)들을 필요로하지 않는다. 그러나 동적스케줄링을 위한 설계 및 구현이 정적스케줄링 방침의 그것보다 어렵고 복잡하여 이에따른 Overhead가 큰것이 일반적이다. 특히 통신(Communication)에 따른 Overhead를 최소화하는등의 문제점이 있다.

3. 시스템 분석을 위한 모형의 정립

어떤 시스템에 대한 하나의 모형을 비형식적인 방법으로 표현하기 위해서는 ① 모형의 구성요소(Component), ② 서술변수(Descriptive Variables), ③ 구성요소들간의 상호작용(Component Interactions)들을 기술하는 것이 필요하다. [1] 구성요소란 모형을 구성하고 있는 제반요소들을 의미하며, 기술변수는 구성요소의 상태를 설명해주는 도구를 말하며, 또한 구성요소들간의 상호작용은 구성요소들의 내부적인 상호작용에 관련된 규칙들을 의미한다.

본 연구에서는 다중프로세서시스템에 관한 분석모형을 설명하기 위하여 상기의 3가지 요인에 해당되는 사항을 다

음과 같이 정렬하였다.

1) 구성요소

PROCESSORS, JOBS, CONTROLLER, MESSAGES

2) 서술변수

PROCESSOR j : P-ID, P-TIME, SERV-JOBS, P-SIGN

JOB i : J-ID, J-CLASS, J-ARRP, J-SERV, J-ARRV-CONT, J-ARRV-PROC

CONTROLLER : TABLE, C-SIGN, FAST-PROC

MESSAGES : ORIGN, DESTN, TEXT

PARAMETERS : Njob, Nprocessor, Nclass, Δt

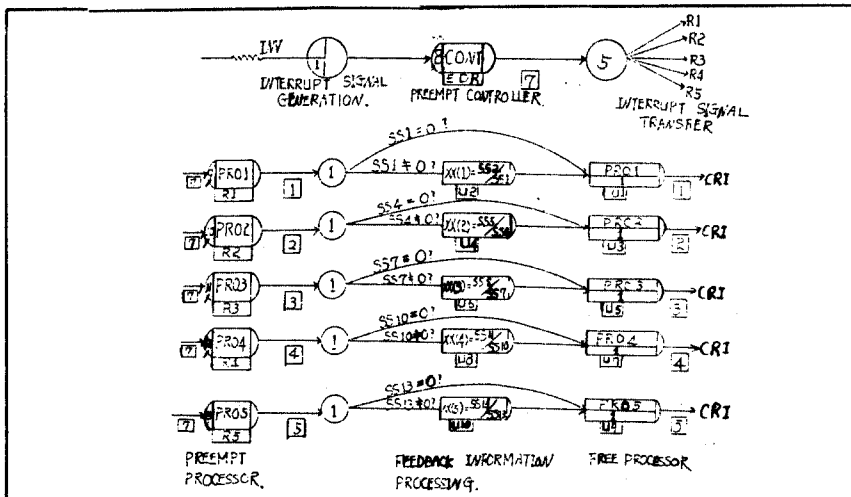
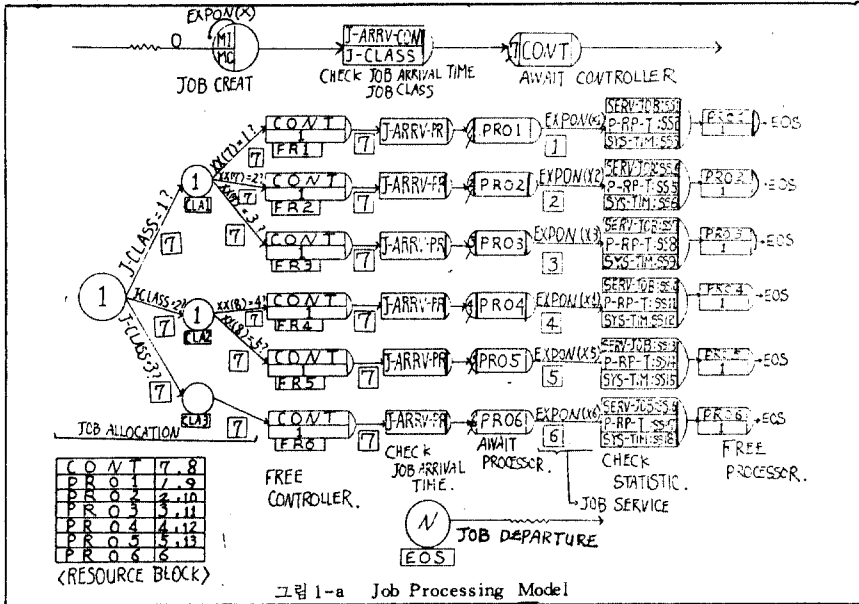
3) 구성요소들간의 상호작용

각 프로세서는 Δt 동안 자신이 처리한 Job 들에 관한 정보를 모았다가 Job Controller 에게 제공한다. Job Con-

troller 는 이 정보를 Table 에 유지시키고 이 Table 의 내용을 참조하여, Job 들을 프로세서에 할당시킨다. Table 의 내용은 Δt 마다 주기적으로 생산되고, 각 프로세서들이 Feedback 정보의 제공을 위하여 처리하고 있는 시간동안에 도착하는 Job 들은 대기상태에 있게되며, 서비스를 받고 있던 Job 들은 Interrupt 에 걸린 상태에 있게 된다.

4. 네트워크 시뮬레이션 모형

본절에서는 앞에서 정립된 시스템분석 모형을 중심으로 다중프로세서시스템의 동적 스케줄링 방침의 적용을 위한 네트워크 시뮬레이션모형을(5) 그림 1에 제시하였다. Job Controller 를 포함한 시스템내의 각 프로세서들을 자원(Resource)으로 하였으며, 자원들을 필요로 하는 Job 들은 엔티티(Entity)로 하였다. 이 모형은 2 가지 처리과정의



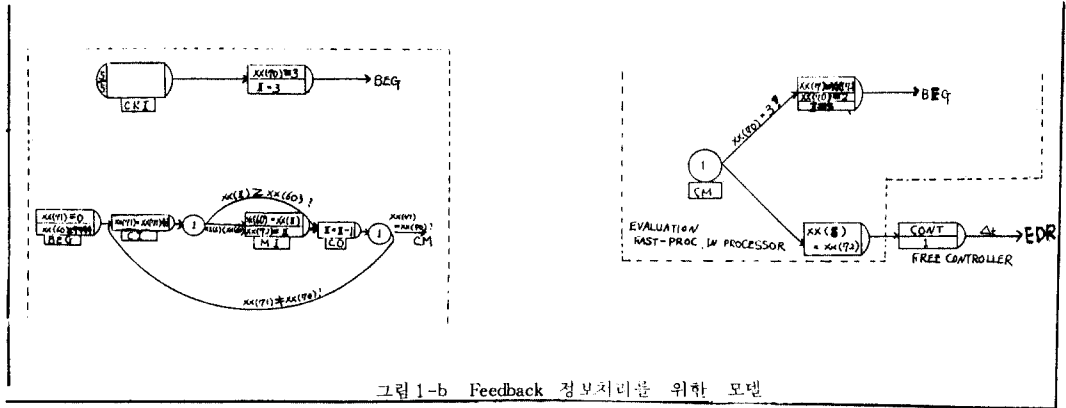


그림 1. 동적스케줄링을 위한 네트워크 시뮬레이션 모델

로 나누어져 있는데, 그림 1-a는 Job의 처리를 위한 것이며, 그림 1-b는 Feedback 정보의 이용을 위한 과정으로서 Δt 마다 주기적으로 프로세서들을 Preemption 시킴으로써 Feedback 정보를 처리하고 있다.

5. 동적스케줄링방침의 최적화

동적스케줄링방침을 위한 여러종류의 시스템 환경하에서 최적 Δt 를 구하기 위해 그림 1의 네트워크 모형을 중심으로 시뮬레이션 언어인 SLAM[6]을 사용하여, Job의 Interarrival time과 프로세서의 Service time이 지수분포를 따른다는 가정하에 시뮬레이션을 수행하였으며, 그 결과들을 그래프와 표로 나타내었다.

본 연구에서 고찰한 시스템의 환경은 다음과 같다.

① One Job Class ($m = 1$)

Five Processors ($n = 5$)

$$J_m = \text{EXPON}(\alpha)$$

$$P_i = \text{EXPON}(\mu_i)$$

$$i = 1, 2, \dots, 5$$

② Two Job Classes ($m = 2$)

Six Processors ($n = 6$)

$$J_m = \text{EXPON}(\beta)$$

$$C_j = \lambda_j, \quad j = 1, \dots, m$$

$$P_i = \text{EXPON}(\mu_i)$$

$$i = 1, 2, 3, \dots, 6$$

③ Three Job Classes ($m = 3$)

Six Processors ($n = 6$)

$$J_m = \text{EXPON}(\gamma)$$

$$C_j = \lambda_j, \quad j = 1, \dots, m$$

$$P_i = \text{EXPON}(\mu_i)$$

$$i = 1, 2, \dots, 6$$

④ Three Job Classes ($m = 3$)

Ten Processors ($n = 10$)

단, J_m : m 개의 Job Class 환경하의 Job Interarrival time
 C_j : j Class Job의 도착비율
 P_i : Processor i 의 Job 서비스 시간
 $\text{EXPON}(X)$: 평균이 X 인 지수분포
 $\alpha, \beta, \gamma, \theta, \mu_i, \lambda_j$ 는 임의의 상수

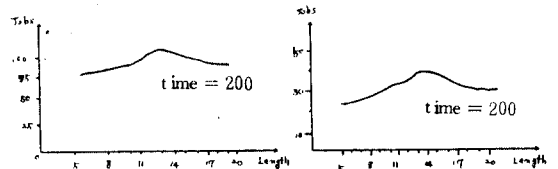
$$J_m = \text{EXPON}(\theta)$$

$$C_j = \lambda_j, \quad j = 1, \dots, m$$

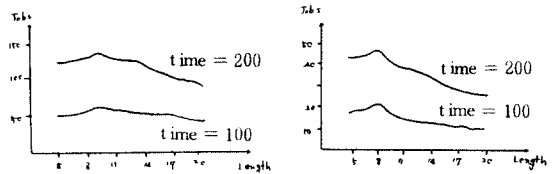
$$P_i = \text{EXPON}(\mu_i)$$

$$i = 1, 2, \dots, 10$$

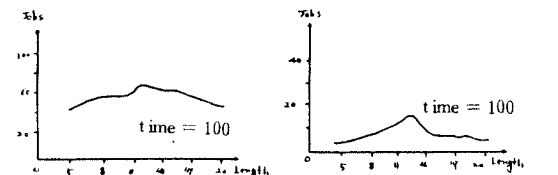
상기의 각 환경에 대한 시뮬레이션 결과는 어느 일정 시점에서 Δt 에 따른 시스템 전체의 Job처리량과, 개별 프로세서의 Job처리량을 아래의 그림 2, 그림 3, 그림 4, 그림 5 및 표 1, 표 2-1, 표 2-2, 표 2-3, 표 2-4, 표 3, 표 4에 제시하였다.



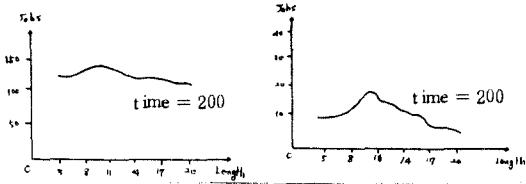
(a) 시스템 평균 Job처리량 (b) 1st Proc.의 Job처리량
그림 2. #C=1, #P=5인 경우



(a) 시스템 평균 Job처리량 (b) 1st Proc.의 Job처리량
그림 3. #C=2, #P=6인 경우



(a) 시스템 평균 Job처리량 (b) 1st Proc.의 Job처리량
그림 4. #C=3, #P=6인 경우



(a) 시스템 평균 Job 처리량 (b) 4th Proc.의 Job 처리량
그림 5. #C=3, #P=10인 경우

표 1 #C=1, #P=5인 경우(각 시뮬레이션의 Job 처리량)

Δt 회수	5	8	11	12	13	14	17	20
1	71	78	81	94	110	99	90	80
2	77	83	87	99	104	97	91	82
3	78	79	86	102	117	108	96	87
4	75	85	83	98	112	104	95	85
평균	75	81	85	98	111	102	93	84

time = 200

단, 회수는 시뮬레이션 회수를 의미함.

표 2-1 #C=2, #P=6인 경우(각 시뮬레이션의 Job 처리량)

Δt 회수	5	7	8	9	11	12	14	17	20
1	65	71	66	72	64	59	57	53	50
2	44	55	56	55	46	45	43	43	39
3	56	65	62	67	59	55	56	50	46
4	43	49	46	46	42	40	41	39	30
5	50	52	53	61	50	48	48	45	42
평균	51	58	56	60	52	49	48	46	42

time = 100

표 2-2 #C=2, #P=6인 경우(각 시뮬레이션의 Job 처리량)

Δt 회수	5	7	8	9	11	12	14	17	20
1	130	135	134	140	135	132	129	116	108
2	122	126	132	130	125	125	116	108	98
3	129	130	133	138	129	126	118	111	101
4	118	125	124	124	122	120	113	102	96
5	121	127	131	133	129	122	117	110	102
평균	124	128	130	134	128	125	119	110	101

time = 200

표 2-3 #C=2, #P=6 (1st Processor Job 처리량)

Δt 회수	5	7	8	9	11	12	14	17	20
1	20	24	26	22	19	16	16	15	14
2	17	18	23	18	16	13	10	9	8
3	18	19	19	20	16	14	12	11	10
4	16	20	19	16	13	12	8	8	10
5	13	16	21	14	10	9	6	5	4
평균	17	19	21	18	15	13	12	11	9

time = 100

표 2-4 #C=2, #P=6 (1st Processor Job 처리량)

Δt 회수	5	7	8	9	11	12	14	17	20
1	51	56	58	54	50	46	42	39	35
2	38	42	43	44	36	34	31	26	24
3	40	48	47	45	40	39	35	29	25
4	43	43	52	48	39	41	36	31	26
5	31	37	40	32	30	29	21	20	18
평균	41	45	48	44	39	37	33	29	26

time = 200

표 3 #C=3, #P=6인 경우(각 시뮬레이션의 Job 처리량)

Δt 회수	5	8	11	12	13	14	17	20
1	41	50	54	65	59	57	53	40
2	42	53	56	68	62	59	56	42
3	43	58	62	73	64	62	59	50
4	46	60	67	74	63	60	57	44
5	43	51	52	63	67	65	60	51
평균	43	55	60	68	63	61	57	44

time = 100

표 4 #C=3, #P=10인 경우(각 시뮬레이션의 Job 처리량)

Δt 회수	5	8	9	10	11	14	17	20
1	119	127	133	138	140	132	123	114
2	121	135	142	142	136	126	117	110
3	126	137	144	147	138	129	122	115
4	120	129	135	144	132	121	115	107
5	117	124	136	145	130	120	113	108
평균	122	130	138	143	136	125	118	111

time = 200

위의 결과로부터 시스템 전체 및 개별 프로세서의 Job 처리량은 Δt 에 따라 점차적으로 변화하고 있음을 알 수 있다. 각 환경에 대한 여러차례의 반복적인 시뮬레이션에서 Job Class 수가 1인 경우에는 $\Delta t = 13$ 일때, Job Class 수가 2이고 Processor 수가 6인 경우에는 $\Delta t = 9$ 일때에 Job 처리량이 최대가 됨을 볼 수 있다. 또한 Job Class 수가 3인 경우에는 Processor 수가 6일때 $\Delta t = 12$ 에서, Processor 수가 10일때 $\Delta t = 10$ 에서 Job 처리량이 가장 많음을 알 수 있다.

각 환경에 대한 최적의 Δt 를 구하기 위하여 시뮬레이션을 수행할때마다 임의의 Random number Seed Value를 줌으로써 Δt 의 변화를 고찰하였다. Seed Value를 임의로 각각 다르게 주었을때의 각 시뮬레이션 결과는 표 2-2의 2번째나 표 3의 5번째 시뮬레이션 결과처럼 미미한 Δt 의 변화를 주고 있다. 또한 시스템의 전체적인

Job 처리량에는 크게 영향을 미치지 않는다.

6. 결론 및 검토

본 논문은 Feedback 정보의 주기적인 이용으로 시스템 상태에 따라 Job 들을 할당하는 동적스케줄링방침의 적용에 있어, 여러 시스템환경에 대한 Feedback Information의 제공시간 간격에 따른 Job 들의 처리량을 비교하였다.

적절한 간격에 의한 feedback 정보의 제공은 시스템의 효율을 높일수 있었다. 그러나, 너무 빈번한 정보의 제공은 시스템의 상태를 보다 명확히 나타낼수 있으나 각 프로세서의 Feedback 정보제공에 따른 Overhead가 증가하므로 시스템의 효율이 저하된다. 반면에 너무 드문 정보의 제공은 시스템상태에 둔감하여 시스템의 효율을 극대화하기 어렵다. 따라서 정보의 제공시간 간격을 결정하기 위한 방법 및 통신지연에 따른 정보의 제공시간 간격에 대한 문제점들을 해결하는 방안이 연구되어야 할것이다.

1. A. ALAN B. PRITSKER, CLAUDE D. PEGDEN, "Introduction to Simulation and SLAM", Wiley New York, 1979.
2. A. K. AGRAWALA, S. K. TRIPATHI, I. G. RICART, "Adaptive Routing using a Virtual Waiting Time Technique", IEEE Trans. Software Eng., Vol. SE-8, pp.76~81, 1982.
3. LIONEL M. NI, KAIHWANG, "Optimal Load Balancing In a Multiple Processor System with many Job-Classes", IEEE Trans. on Software Eng., Vol. SE-11, No.5 pp.491-496, May, 1985.
4. Y. C. CHOW, W. H. KOHLER, "Models for Dynamic Load Balancing In a Heterogeneous Multiple Processor System", IEEE Trans. Comput., Vol. C-28, pp.345~361, May, 1979.
5. ZEIGLER, B. P., "Theory of Modelling and Simulation", Wiley New York, 1976.
6. 오종진, 백두권, 조용환, "다중프로세서시스템의 동적부하균등화에 관한 연구", 한국정보과학회, 86 가을 학술발표논문집 제 13 권 2 호, pp. 87 ~ 92, 1986, 10.