

멀티프로세서 시스템을 위한 동적 전압 조절 기반의 효율적인 스케줄링 기법

論 文
57-3-14

An Efficient Scheduling Method based on Dynamic Voltage Scaling for Multiprocessor System

盧 暉 佑* · 朴 彰 雨* · 金 錫 潤†
(Kyung-Woo Noh · Chang-Woo Park · Seok-Yoon Kim)

Abstract - The DVS(Dynamic Voltage Scaling) technique is the method to reduce the dynamic energy consumption. As using slack times, it extends the execution time of the big load operations by changing the frequency and the voltage of variable voltage processors. Researches, that controlling the energy consumption of the processors and the data transmission among processors by controlling the bandwidth to reduce the energy consumption of the entire system, have been going on. Since operations in multiprocessor systems have the data dependency between processors, however, the DVS techniques devised for single processors are not suitable to improve the energy efficiency of multiprocessor systems. We propose the new scheduling algorithm based on DVS for increasing energy efficiency of multiprocessor systems. The proposed DVS algorithm can improve the energy efficiency of the entire system because it controls frequency and voltages having the data dependency among processors.

Key Words : Dynamic Voltage Scaling, Multiprocessor, Task Scheduling

1. 서 론

최근 휴대용 단말기의 급속한 발전에 따라 단말기의 크기는 점점 작아지고 있지만, 단말기의 기능은 비디오, 전화, 오디오, 카메라, 디지털 멀티미디어 방송 등 많은 종류의 멀티미디어 응용을 포함하여 복잡해지고 있으며 이런 다양한 기능을 단일 프로세서 시스템으로 수행하는 것은 한계가 있기 때문에 멀티프로세서 시스템이 요구되고 있다. 또한 멀티미디어의 지원 등 다양한 부가 기능을 탑재하는데 다른 저전력 소비 관리가 필수적이기 때문에 회로 수준에서 운영체제, 미들웨어, 응용 수준에 이르기까지 다양한 추상화 단계의 효율적인 저전력 설계 기법들이 제안되었다[1].

동적 전압 조절 (DVS: dynamic voltage scaling)은 마이크로프로세서 기반의 시스템에서 가장 주목받는 기술 중의 하나이다[2]. 이 기법은 작업이 없는 유휴 시간을 활용하여 전압 및 동작 주파수를 낮춤(DFS: dynamic frequency scaling)으로써 시스템의 에너지 효율성을 높이는 것이며 이미 Transmeta의 Crusoe와 AMD의 이동단말기 프로세서, Intel의 Xscale 등은 전압 조절기능을 가진 프로세서들로 널리 알려져 있다. 그리고 프로세서뿐만 아니라 주파수 조절 가능한 병렬 링크와 직렬 링크도 제안되었다[3, 4, 5]. 링크의 데이터 전송 에너지를 줄이기 위하여 데이터 전송 시간 제약을 넘지 않는 범위 내에서 주파수와 전압을 낮춤으로써

대역폭을 조절한다.

하지만 지금까지 제안된 동적 전압 조절 기법[6]은 단일 프로세서 시스템을 모델로 하여 제안되었으며 멀티프로세서 시스템을 대상으로 한 연구는 진행이 미비하였다. 최근에 들어서야 일부 시작되고 있으나 기존의 단일 프로세서 칩에서 사용되던 기법들을 단순히 확대 적용하는 차원에 연구가 진행되고 있다. 그렇기 때문에, 프로세서간의 데이터 의존성이 높은 멀티프로세서 시스템에 적용을 할 수 있는 동적 전압 조절 기법이 필요하게 되었다.

본 논문에서는 기존에 제안된 동적 전압 조절과 동일 시간대에 데이터 의존성이 없는 작업들의 에너지 감소율을 이용하여 멀티프로세서 시스템에 적용할 수 있는 스케줄링 알고리즘을 제안한다. 먼저, 알고리즘을 구현하는 과정에서 작업의 우선순위를 정하기 위하여 주어진 작업들의 에너지 감소율을 정의하였으며, 동일 시간대에 데이터 의존성이 없는 작업들의 에너지 감소율 합과 크기를 우선순위로 사용하여 동적 전압 조절을 적용하였다. 그리고 시스템 환경은 이종 프로세서들과 프로세서들을 연결하고 있는 링크들로 구성되어 있으며, 시스템 구조와 위상은 결정되어져 있다고 가정하였다. 또한 실제 시스템과 유사한 환경을 구축하기 위하여 작업의 모형을 단순한 데이터 흐름(data flow)으로만 보지 않고 제어 흐름(control flow)까지 고려하였다.

서론에 이어, 2장에서는 자동 전압 조절 기법을 적용한 멀티프로세서 시스템의 입력된 작업의 모형을 보여주고 3장에서는 소모 에너지와 작업 실행 시간의 상관관계를 실제 전압 조절 가능한 프로세서에 맞추어 소개한다. 4장에서는 멀티프로세서 시스템에서 데이터 의존성을 고려한 새로운 동적 전압 조절 알고리즘을 소개하고, 제안한 알고리즘과 기존에 제안된 멀티프로세서 시스템을 위한 동적 전압 조절

* 學生會員 : 崇實大學 컴퓨터學科 碩士課程
† 교신저자, 正會員 : 崇實大學 컴퓨터學科 教授 · 工博
E-mail : ksy@ssu.ac.kr
接受日字 : 2007年 12月 27日
最終完了 : 2008年 1月 22日

기법의 차이를 보여준다. 5장에서는 제안한 알고리즘의 효율성을 실험한 결과를 보여주며, 6장에서는 결론을 맺는다.

2. 입력된 작업의 모형화

입력된 작업을 모형화 하기 위하여, 멀티프로세서와 멀티프로세서를 연결하고 있는 링크들의 위상은 정해져 있다고 가정 하고 이전 연구에서 흔히 사용하는 DAG(directed acyclic graph)를 이용한다[7].

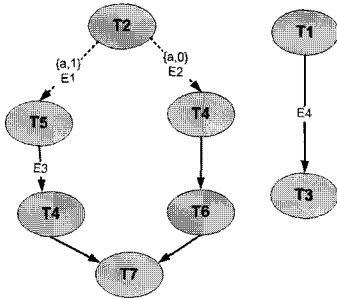


그림 1 입력된 작업의 모형화
Fig. 1 The modeling of input tasks

그림 1은 전압 조절 가능한 멀티프로세서에 입력된 작업들을 모형화 한 그림이다. DAG에는 노드들과 연결선으로 구성이 되어 있다. 노드들은 작업들을 표현하고 있으며 연결선들 멀티프로세서의 데이터가 전송되는 링크들을 표현한 것이다. 연결선은 데이터의 의존성만을 나타내는 실선과 데이터 의존성과 조건에 따른 분기를 동시에 나타내는 점선으로 이루어져 있다. 즉, 그림 1의 T2는 변수 a의 값이 1이면 링크 E1을 통해 T5를 수행하게 되고, 반대로 a의 값이 0이면 링크 E2를 통해 T4를 수행하게 된다. 그리고 T4와 T6을 연결하는 연결선처럼 연결선에 이름이 없는 경우는 동일 프로세서 내에서 수행되는 작업이다.

표 1 작업들의 구성 요소 예

Table 1 The example of elements of tasks

Task #	Probability	Delay	Power
1	1.0	5.2	2.0
2	1.0	4.0	1.5
3	1.0	5.2	2.0
4	1.0	3.0	1.0
5	0.5	2.5	1.5
6	0.5	2.0	2.0
7	1.0	8.0	4.0

표 1과 표 2는 각각 작업들과 링크들을 구성하는 튜플들의 요소를 나타내고 있다. 작업들은 표 1과 표 2처럼 (probability, delay, power)로 구성이 되며, 링크들은 (probability, message size)로 구성된다[7]. 전압 조절 가능한 프로세서와 주파수 조절 가능한 링크내에서의 작업들과 링크는 전압과 주파수 값에 따라 다양한 상태를 가질 수 있

으며 이때 구성 요소의 값들도 바뀌게 된다. 즉, 작업들과 링크들은 다른 실행 시간동안 수행될 수 있으며, 그에 따라 에너지 소비도 다르게 된다.

표 2 링크들의 구성 요소 예

Table 2 The example of elements of links

Edge #	Probability	Message size
1	1.0	50000
2	0.5	35000
3	0.5	50000
4	1.0	28000

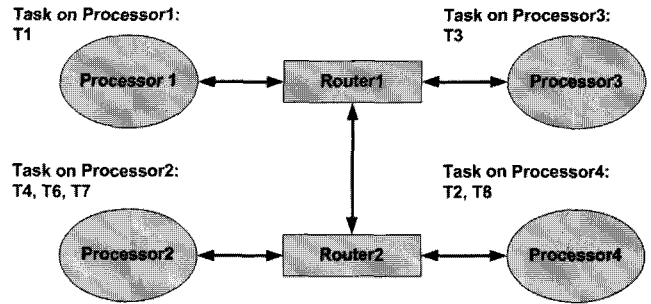


그림 2 전압 조절 가능한 멀티프로세서의 모형
Fig. 2 The modeling of the multiprocessor that controls voltage

그림 2는 전압 조절 가능한 멀티프로세서와 프로세서와 프로세서 사이의 데이터를 전송하는 링크들의 모형을 나타내고 있다. 실제 링크들은 [8]에서 확인해볼 수 있는 것과 같이 다양한 네트워크 구조를 가지고 있으나, 본 논문에서는 동적 전압 조절 기법을 사용하기 이전에 링크 패스는 이미 정해졌다고 가정한다.

3. 소모 에너지와 작업 실행 시간

본 장에서는 전압 조절 가능한 프로세서와 링크의 에너지 소모와 작업 실행 시간의 상관관계를 보여준다. 먼저 공급 전압과 주파수와의 관계를 살펴 본 후 주파수와 공급전압이 동적 에너지 소비에 어떤 영향을 미치는지 살펴본다. 그리고 실제 사용되고 있는 인텔의 Xscale 프로세서의 조건에 맞추어 동적 소비 에너지와 작업 실행 시간의 상관관계를 분석한 뒤 앞서 연구되어 왔던 주파수 조절 가능한 링크 구조에서 송신단에 입력되는 주파수 크기 변화와 비트 당 보내는데 필요한 에너지 소비의 상관관계를 보여준다.

3.1 프로세서의 동적 에너지 소비와 작업 실행 시간

동적 에너지 소비는 시스템의 총 에너지 소비의 큰 부분을 차지하고 있다. 동적 전력 소비를 결정하는 요소로는 공급전압, 클럭 주파수, 신호의 천이 횟수, 전기 용량이 된다. 수식 (1)은 동적 전력 소비를 나타내고 있다.

$$p = 1/2fNCV_{dd}^2 \tag{1}$$

$$f = k(V_{dd} - V_t)^\alpha / V_{dd} \quad (2)$$

여기서 p 는 동적 전력 소비, f 는 클럭 주파수, N 은 신호의 천이 횟수, C 는 전기 용량, V_{dd} 는 공급전압을 의미한다[9]. 수식 (1)을 통해 알 수 있듯이 프로세서가 소모하는 동적 전력 소비는 프로세서에 공급되는 전압 값의 지수 함수에 비례하여 증가한다. 수식 (2)는 공급 전압과 문턱 전압 (V_t)의 변화에 따른 클럭 주파수를 구하는 식이다. k 는 상수이며 α 는 1과 2사이의 값을 가진다. 수식 (2)를 통해 프로세서의 실행 속도는 공급되는 전압 값에 비례함을 알 수 있다.

그림 3은 수식(1)과 (2)를 이용하여 공급 전압을 1.6V에서 0.85V까지 변할 때, 주파수의 변화를 보여주고 있다.

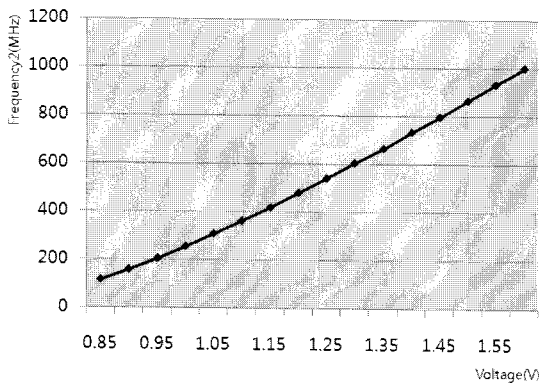


그림 3 공급 전압 변화에 따른 주파수 변화
Fig. 3 The frequency change depending on supply voltage change

에너지 동적 에너지 소비(E)는 수식 (3)과 같이 동적 전력 소비에 시간(t)를 곱해줌으로써 구할 수 있으며, 이때 시간(t)는 수식 (4)와 같이 클럭 사이클 개수(n)와 클럭 주파수로 표현이 가능하다.

$$E = t \times p = t \times (1/2fNCV_{dd}^2) \quad (3)$$

$$t = n/f \quad (4)$$

위의 수식 (2)와 (3), (4)에서 볼 수 있듯이 소비되는 에너지는 시간에 비례하여 증가함을 알 수 있지만, 만일 공급전압이 변하게 되면 에너지 소비는 단순히 시간에 비례해서 증가하지 않고, 시간과 공급전압의 제곱근의 곱에 비례함을 알 수 있다. 즉, 선형적으로 증가하지 않는다. 그림 4는 수식 (1)과 (2), (3)을 이용하여 공급 전압 변화에 따른 소비되는 에너지를 나타내고 있다. 실제 인텔의 Xscale 프로세서의 변수와 비슷한 조건인 최대 주파수(f_{max})는 1GHz, 최대 공급 전압(V_{max})은 1.6V, 최소 공급 전압(V_{min})은 0.85V, α 값은 2, 문턱전압(V_t)는 0.6V를 사용하였다. 그림 4에서 보는 바와 같이, 에너지 소비는 신호의 천이 횟수에 비례함을 알 수 있다.

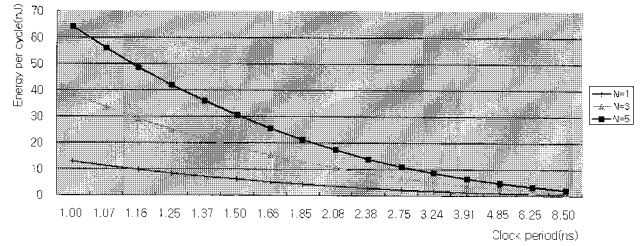


그림 4 공급 전압 변화에 따른 한 사이클에 의해 소비되는 에너지
Fig. 4 The consumption energy per cycle depending on supply voltage change

3.2 링크에서의 데이터 전송 에너지 소비와 데이터 전송 시간

그림 5는 앞서 연구되어 왔던 주파수 조절 가능한 링크구조에서 데이터 전송에 소비되는 에너지와 데이터 전송 시간의 상관관계를 보이고 있다[7]. 이때 사용되는 공급 전압 범위는 0.9V부터 2.5V까지이며 공급 전압에 상응하는 전송 속도는 450Mb/s에서 3.7Gb/s 이고 90MHz에서 740MHz까지의 주파수 범위를 가진다[10].

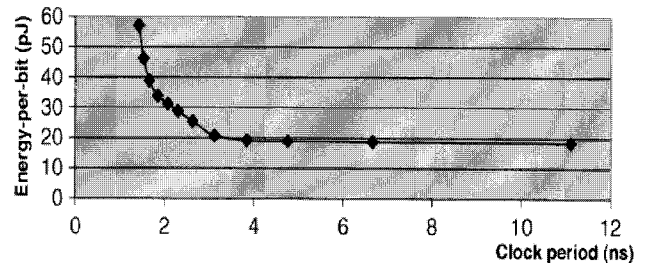


그림 5 주파수 변화에 따른 한 사이클에 의해 소비되는 에너지[7]
Fig. 5 The consumption energy per cycle depending on frequency[7]

그림 5에서 볼 수 있듯이, 주파수 조절 가능한 링크의 비트 당 에너지 소비 그래프는 그림 4의 공급 전압 변화에 따른 소비되는 에너지 그래프와 유사한 형태를 띠고 있다.

4. 다중 프로세서 환경에서 동적 전압 조절을 이용한 스케줄링 기법

동적 전압 조절을 이용하려면 작업에 우선순위를 두어 유희시간을 우선적으로 활용할 수 있게 하여야 한다. 단일 프로세서 시스템에서는 EDF(earliest deadline first), RM(rate-monotonic)등이 사용되었으나[11] 멀티프로세서 시스템에서는 작업들이 병렬적으로 처리되기 때문에 단일 프로세서 시스템에서 사용하는 방법을 사용할 수 없다. 본 장에서는 프로세서에 작업이 들어왔을 때, 동적 전압 조절을 할 작업의 우선순위를 정하기 위한 요소로 사용하게 될 에너지 감소율(ERR: energy reduction rate)과 평균 에너지 감소율

(AERR: average energy reduction rate)을 소개한다. 그리고 기존에 제안되어진 멀티프로세서 환경에서의 동적 전압 조절 기법과 본 논문에서 제안하는 동적 전압 조절 기법을 소개한 뒤, 차이점을 보여준다.

4.1 에너지 감소율

에너지 감소율은 전압(혹은 주파수)을 조절하기 전에 소비되는 에너지와 조절한 후의 소비되는 에너지 차이를 실행 시간의 차이로 나누는 것으로 정의한다. 수식 (5)는 에너지 감소율을 표현한 것이다.

$$ERR(v) = - \frac{\Delta E(v)}{\Delta t} \quad (5)$$

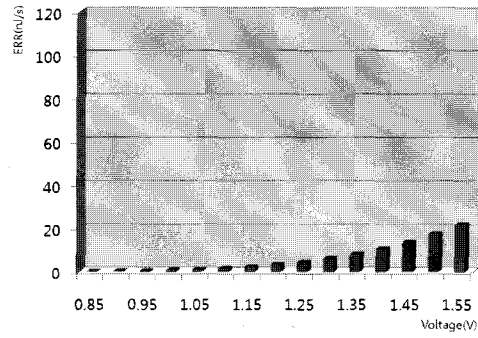
에너지 감소율은 주파수를 낮추게 될수록 현저히 감소하게 된다. 예를 들어 그림 4의 스위칭 횟수가 5인 작업에서 프로세서의 클럭 주파수가 1GHz로 작동할 때, 한 사이클 당 동적 에너지 소비는 64nJ이다. 만일 이 작업을 수행할 때 공급 전압을 조절하여 주파수를 600MHz로 낮추었을 경우, 한 사이클 당 동적 에너지 소비는 25.48nJ로 에너지 측면에서 얻은 효과는 38.52nJ이 된다. 반면에 전압을 낮추기 이전의 주파수가 600MHz라고 가정하고 전압을 낮추어 프로세서의 클럭 주파수가 200MHz로 동작하게 되면, 이때 동적 에너지 소비는 4.65nJ이 되어 에너지 측면에서 얻은 효과는 20.38nJ이 된다. 즉, 같은 크기만큼 주파수를 감소 시켰지만 동적 에너지 소비의 효율 면에서는 최대한 높은 주파수를 낮은 주파수로 낮추는 것이 좋다.

그림 6의 (a)와 (b), (c)는 각각 그림 4의 N=1, N=3, N=5인 작업을 기준으로 하여, 공급 전압을 1.6V에서 0.05V씩 낮추어가면서 0.85V까지의 프로세서의 초당 에너지 소비를 나타낸 것이다. 그림 6에서 볼 수 있듯이 전압을 낮출수록 에너지 감소율은 작아지게 된다.

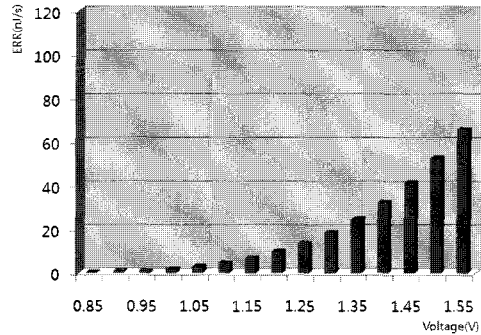
$$AERR(v_i) = probability_i \cdot ERR(v_i) \quad (6)$$

수식(6)은 특정 작업 *i*의 전압을 *v*로 변화 시켰을 때의 평균 에너지 감소율(average energy reduction rate)를 나타내고 있다. 분기로 인해 DAG에서 나누어지는 작업들의 수행될 확률은 각각 다르게 되기 때문에, 작업의 에너지 감소율에 작업이 일어날 확률을 가중치를 줌으로써 평균 에너지 감소율을 구할 수 있다.

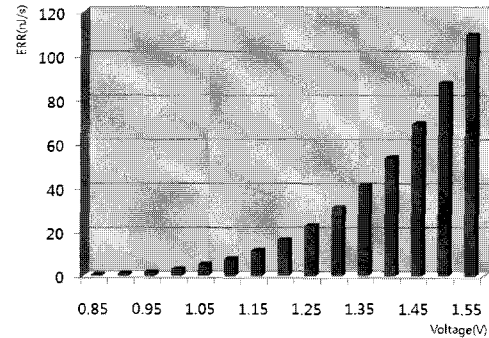
단일 프로세서 시스템에서는 에너지 감소율이 큰 작업을 우선으로 하여, 전체 작업의 유휴시간을 이용하여 스케줄링을 하면 되지만, 멀티프로세서 시스템에서 작업들이 병렬적으로 진행되기 때문에 단순히 에너지 감소율이 큰 작업이 우선으로 유휴시간을 이용하여 스케줄링을 할 경우 다른 작업에 영향을 주게 되어, 시스템 전체 성능에 큰 영향을 미칠 수가 있다. 그렇기 때문에 멀티프로세서 시스템에서는 유휴시간을 최대한 활용할 뿐만 아니라, 다른 프로세서의 작업들에게 끼치는 영향까지도 고려하여야 한다. 이에 관한 기법은 뒤에서 설명하기로 한다.



(a) N=1



(b) N=3



(c) N=5

그림 6 공급 전압 변화에 따른 에너지 감소율

Fig. 6 The energy reduction rate depending on supply voltage change

4.2 멀티프로세서 환경에서의 동적 전압 조절 알고리즘

이전 연구된 멀티프로세서 환경에서의 동적 전압 조절 기법[7]은 에너지 감소율을 이용하여 일반적인 단일 프로세서의 동적 전압 조절 기법을 단순히 확대해서 적용시키고 있다. 먼저 EST(earliest start time)과 LFT(latest finish time)을 정의 한다. 특정 작업 *i*의 EST와 LFT는 (7)과 (8)처럼 정의할 수 있다.

$$est_i = \max_{j=predecessors} (EST_j + wst_delay_j) \quad (7)$$

$$lft_i = \min(d_i, \min_{j \in \text{successors}} (LFT_j - wst_delay_j)) \quad (8)$$

수식(7)과 (8)에서 successors와 predecessors는 DAG에서 자식 노드와 부모 노드를 각각 의미하고 있다. d_i 는 작업 i 의 데드라인, wst_delay 는 동적 전압 조절을 하여 최대 수행 시간을 의미한다.

이전에 제안된 멀티프로세서 환경에서 동적 전압 조절 기법[7]에서는 제일 먼저 작업간의 유희시간이 존재하고 특정 작업에 동적 전압 조절 기법을 적용할 수 있는지를 검사한다. 만일 동적 전압 조절 기법을 적용할 작업이 있다면 작업을 DAG처럼 위치적으로 정렬을 한 뒤, 우선적으로 동적 전압 조절을 할 작업들을 선택한다. 이때, 가장 AERR이 큰 작업들이 선택이 되고, 이 작업들은 (7)과 (8)의 조건에 맞추어 동적 전압 조절을 하게 된다. 그리고 reference event가 존재하여 worst delay mode로 전압 조절이 가능한 작업은 수행 완료 시간을 reference event에 맞추어 늦출 수 있다. 그림 7은 이런 reference event가 있는 작업이 있을 때, 동작 원리를 보여주고 있다. 그림 7의 T1과 T2는 앞선 작업의 분기로 인해 나누어지고 있으며 T1과 T2가 종료되면 T3이 수행된다. 이때 T2가 T1보다 작업이 빨리 끝나는 경우, 스케줄러는 T2를 T1의 데드라인까지 수행시간을 늦추어 동작을 해도 전체 시스템에 아무런 영향을 끼치지 않기 때문에 T2가 수행될 때의 전압을 낮추어 에너지 소비를 줄일 수 있다.

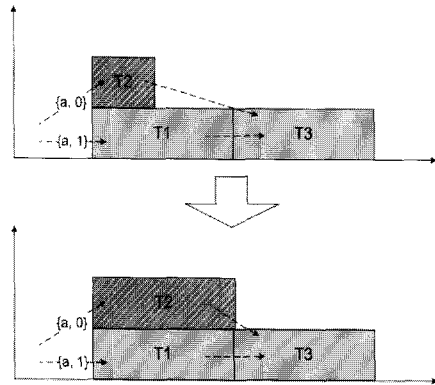


그림 7 reference event의 worst delay mode로 전압조절
Fig. 7 The voltage scaling on worst delay mode of reference event

위와 같은 방법을 계속 반복적으로 수행함으로써 전체 동적 에너지 소비를 줄일 수 있다. 그림 9는 [7]에서 제시한 동적 전압 조절 기법으로 입력 작업이 그림 8과 같을 때의 스케줄링 결과를 보여주고 있으며 프로세서 구조는 그림 2와 같이 구성되어 있다고 가정하였다.

그림 9에서 T2와 T8의 ERR이 가장 높고, T4와 T7이 다음으로 높은 ERR을 가진다고 가정하면, T2와 T8이 (7)과 (8)의 조건을 만족 하는 범위 내에서 스케줄링을 하게 된다. T4와 T7은 같은 ERR을 가지지만 그림 8의 DAG에서 보는 바와 같이 T4는 분기로 인해 수행되는 작업이다. 즉, T4가 수행될 확률은 T7보다 작기 때문에 T4의 AERR은 T7의 AERR보다 작다. 스케줄러는 T7의 수행시간을 유희시간을

활용하여 스케줄링 한다. 그림 8의 마지막 그림이 최종적으로 스케줄링 된 작업들의 모형이다. 하지만 멀티프로세서 환경에서 에너지 소비를 가장 효율적으로 하기위한 스케줄링 방법은 NP-complete 문제이다. 즉, 프로세서와 수행하는 작업 수가 늘어나면 일반적인 다항식으로는 가장 효율적인 스케줄링 방법을 찾을 수가 없다. 이전 연구[7]에서 제안한 방법을 사용하면 스케줄링 하기전의 에너지 소비보다는 효율적이지만 이 또한 다른 스케줄링 결과보다 효율적인 방법이라고 할 수 없다.

이런 NP-complete 문제를 효율적으로 해결하기 위해 제안된 여러 방법 중 효율적으로 널리 알려진 A*(A-star) 알고리즘[12]을 이용하면 효율적인 스케줄링 결과를 얻을 수 있다. 수식 (9)는 본 논문에서 제안하는 스케줄링을 하기 위해 각 작업에 주어지는 우선순위를 A* 알고리즘을 이용하여 표현하고 있다.

$$P_i = \sum_{j \in \text{runtask}(t_{eval})} AERR(v^j) + AERR(v^i) \quad (9)$$

where $i \in \text{runtask}(t_{eval})$

수식 (9)의 $\text{runtask}(t)$ 는 시간(t)가 작업시작시간 이상이고 작업종료시간 이하일 때, 이를 만족하는 작업들을 의미하며, 작업의 평가시점(t_{eval})은 그림 10과 같이 각 작업이 스케줄링 되기 전의 시작시간을 나타내고 있다.

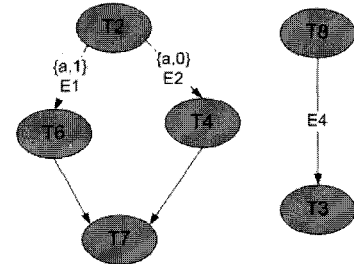


그림 8 입력 작업 모형화 예
Fig. 8 The example of the modeling of input tasks

먼저 runtask 들의 AERR 총 합이 가장 높은 평가시점에서, AERR이 가장 높은 작업이 가장 높은 우선권을 가지고 스케줄링 하게 된다. 그리고 그림 7과 같이 reference event가 존재하여 worst delay mode로 전압 조절이 가능한 작업은 수행 완료 시간을 reference event에 맞추어 늦출 수 있다. 이와 같은 작업을 반복적으로 하여 더 이상의 유희시간이 존재하지 않으면 동적 전압 조절은 완료되게 된다. 그림 11은 이전 논문[7]에서 제안한 방법과 본 논문에서 제안하고 있는 방법의 차이를 보여주고 있다.

그림 11의 위의 그림은 4개의 작업으로 이루어져 있으며, 3개의 작업은 앞선 1개의 작업에 의존적이며 각 작업의 AERR이 위의 그림처럼 계산이 되었다고 가정한다. 앞선 연구에서 제안한 방법을 사용하면, 그림 (a)처럼 가장 큰 AERR을 가진 작업을 우선적으로 유희시간을 활용하게 된다. 그림 (b)는 본 논문에서 제안한 방법으로 runtask 들의 총 AERR의 합을 먼저 계산한 뒤, runtask 중 가장 AERR

이 높은 작업부터 우선적으로 유휴시간을 활용하여 스케줄링을 한 결과이다. 그림에서 볼 수 있듯이 본 논문에서 제안한 기법 결과가 이전 연구보다 효율적임을 알 수 있다. 그림 12는 제안하는 알고리즘의 의사코드이다.

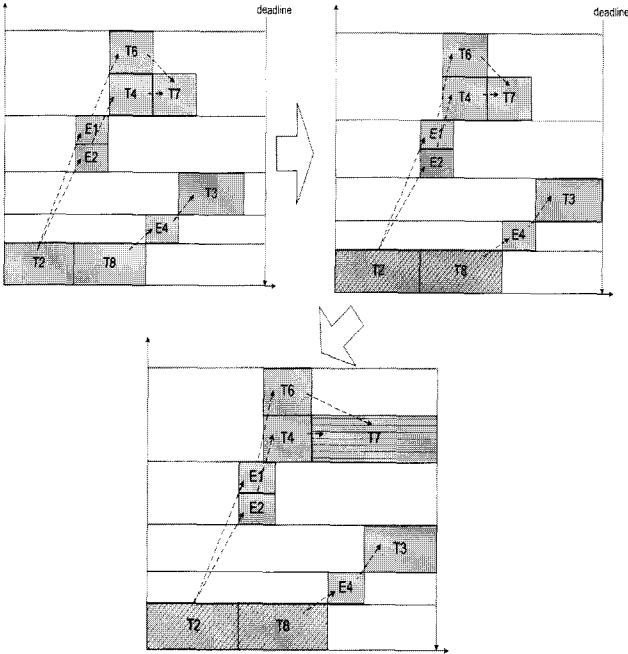


그림 9 [7]에서 제안하는 동적 전압 조절 기법을 이용한 스케줄링 결과

Fig. 9 The results of proposed dynamic voltage scaling method in [7]

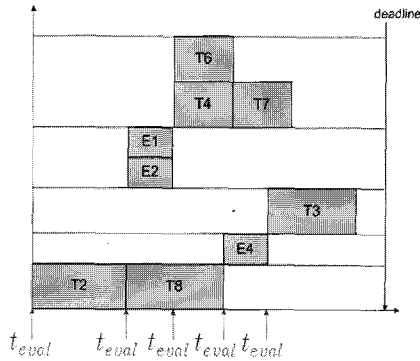


그림 10 전체 작업에서의 각 평가시점(t_{eval})

Fig. 10 Evaluation points on entire tasks

5. 실험결과

본 장에서는 알고리즘을 구현한 응용 프로그램을 이용하여 멀티프로세서 시스템 환경을 구축하고, 10개의 작업들을 무작위로 만들어 각 프로세서에 해당 작업들을 할당한 뒤 동적 전압 조절 기법을 사용하지 않았을 때를 기준으로 AERR의 크기를 우선순위로 사용한 동적 전압 조절 기법[7]과 앞서 제안한 A* 알고리즘을 이용하여 계산되어진 우선

순위를 이용하여 동적 전압 조절 기법을 적용한 결과를 비교 분석한다.

실험 결과 분석의 용이성을 위하여 멀티프로세서 시스템 플랫폼은 6개의 프로세서와 그림 2에서와 같이 간단한 링크 구조를 이용하여 설계하였으며, 6개의 프로세서는 동일한 프로세서들로 구성되어 있다고 가정한다.

그림 13은 AERR의 크기를 우선순위로 이용하여 동적 전압 조절 기법을 적용한 방법과, 제안한 A* 알고리즘을 이용하여 계산되어진 우선순위를 이용하여 동적 전압 조절 기법을 적용한 방법의 유휴시간 활용률을 보여주고 있다.

그림 13에서 보는바와 같이 작업들의 유휴시간을 활용하는 측면에서는 거의 동일하게 90% 이상의 효율성을 보이고 있으며 AERR의 크기를 우선순위로 사용하여 동적 전압 조절 기법을 적용한 방법이 A* 알고리즘을 이용하여 계산되어진 우선순위를 이용하여 동적 전압 조절 기법을 적용한 방법보다 약간의 효율성이 높음을 알 수 있다.

그림 14는 주어진 작업들이 실행될 때, 소비되는 에너지의 크기를 동적 전압 조절을 하지 않았을 때를 기준으로 상대적인 효율성을 보여주고 있다. A*알고리즘을 이용하여 동적 전압 조절 기법을 적용하였을 때의 상대적인 에너지 효율성은 평균 39.50%, AERR 크기를 우선순위로 사용하여 동적 전압 조절 기법을 적용하였을 때의 상대적인 에너지 효율성은 평균 28.56%로 나타났다.

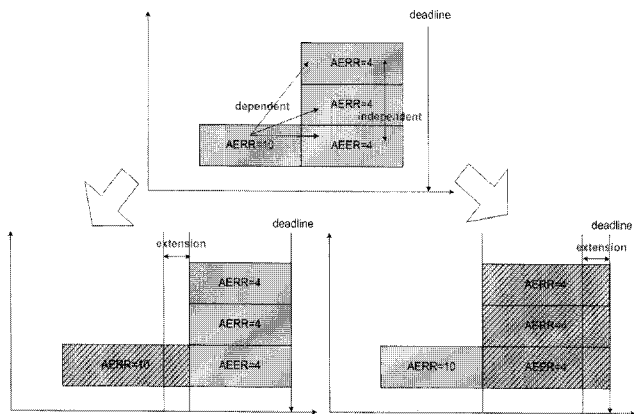
그림 14의 결과와 같이 에너지 효율적인 측면에서는 A* 알고리즘을 이용하여 동적 전압 조절 기법을 적용하였을 때, AERR의 크기를 우선순위로 사용하여 동적 전압 조절 기법을 적용한 방법보다 효과적이라고 볼 수 있다. 그림 14의 3, 4, 5번째 작업들은 A* 알고리즘을 이용하여 스케줄링을 했을 경우, 다른 작업들에 비해 상대적으로 높은 효율성이 높게 나타나는 것을 확인할 수 있다. 이는 3, 4, 5번의 작업들이 각 해당 프로세서에 작업이 나뉘어서 할당 될 때, 평가시점(t_{eval})에 서로 영향을 주지 않는 작업들이 많이 있다는 것을 나타내고 있다. 이와 반대로 그림 14의 6, 10번째 작업들은 평가시점(t_{eval})에 서로 영향을 주지 않는 작업들의 수가 적기 때문에, A* 알고리즘을 이용하여 스케줄링 했을 때와 AERR의 크기를 우선순위로 사용하여 스케줄링 했을 때의 에너지 효율성은 거의 유사하게 나타났다.

6. 결 론

본 논문에서는 멀티프로세서 시스템에서 멀티프로세서 시스템의 에너지 효율을 높이기 위하여 기존에 제안된 동적 전압 조절 기법과 동일 시간대에 데이터 의존성이 없는 작업들의 에너지 감소율을 이용하여 멀티프로세서 시스템에 적용할 수 있는 스케줄링 알고리즘을 제안하였다. 먼저, 기존에 연구된 단일 프로세서 내에서의 동적 전압 조절 기법을 소개 하였으며, 작업 실행 시 소모되는 동적 에너지 소비와 작업 실행 시간의 상관관계를 보였다. 그리고 멀티프로세서 시스템 환경에서 A* 알고리즘을 이용하여 기존에 연구되어 왔던 단순히 AERR의 크기를 우선순위로 하여 동적 전압 조절을 하지 않고, 평가시점(t_{eval})에 의존적이지 않는 작업들의 AERR의 합과 각 작업들의 AERR의 크기를 우선

순위를 결정하는 요소로 사용을 하였다.

본 알고리즘을 C++로 구현하여 실험을 한 결과, 본 논문에서 제안하는 A* 알고리즘을 이용하여 동적 전압 조절 기법을 적용하였을 때의 에너지 효율성은 동적 전압 조절 기법을 사용하지 않았을 때보다 39.50%의 에너지 효율성이 증가하였고, 단순히 AERR의 크기만을 이용하여 동적 전압 조절 기법을 적용하였을 때의 에너지 효율성은 28.56%로 나타났다. 이 수치는 입력된 작업의 의존성과 병렬성 그리고 남은 유휴시간의 크기에 따라 달라질 수 있지만, 기존에 제안되었던 방법보다 에너지 소비 측면에서 효율적임을 알 수 있다.



(a) AERR이 큰 작업을 기준 (b) 의존성이 없는 작업들의
으로 동적 전압 조절 기 AERR의 합을 기준으로
법 적용 동적 전압 조절 기법 적용

그림 11 [7]에서 제안한 방법과 본 논문에서 제안한 방법의
결과 비교

Fig. 11 The comparison of results between the proposed
method in [7] and in this paper

```

1. for all task ∈ taskSet do
2.   timeEvalList ← taskStartTime
3. end for
4. until extensible task
5.   for all time ∈ timeEvalList do
6.     if(sum of runtask(time)'s AERR is
maximum)
7.       dvsScheduling(task ∈ runtask(time)
&& task's AERR is maximum)
8.     end if
9.   end for
10. end until
11. update task's start time and finish time
    
```

그림 12 A* 알고리즘을 이용한 스케줄링 의사코드
Fig. 12 The scheduling pseudocode using A* algorithm

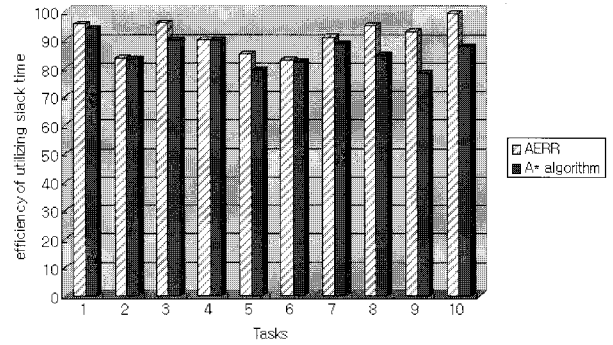


그림 13 작업들의 유휴시간 활용률
Fig. 13 The using rate of idle time of tasks

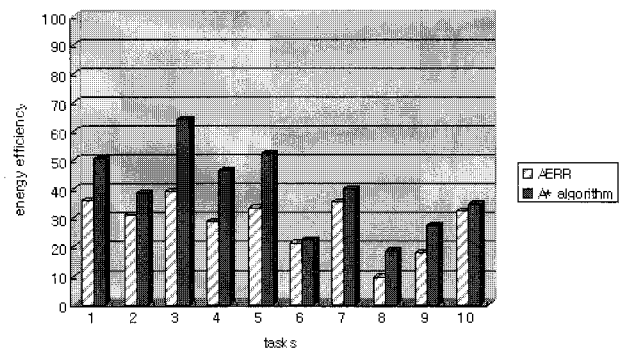


그림 14 소모 에너지 효율성
Fig. 14 The consumption energy efficiency

감사의 글

본 연구는 한국과학재단 특정기초연구(No. R01-2005-000-11215-0(2005))와 숭실대학교 교내연구비의 지원으로 이루어졌습니다.

참고 문헌

[1] M. Pedram and J. M. Rabaey, *Power Aware Design Methodologies*, Kluwer Academic, 2002.
 [2] A. P.Chandrakasan, A. P. Sheng, and R. W. Brodersen, "Low-Power CMOS Digital Design," *IEEE Journal of Solid-State Circuits*, Vol. 27, No. 4, pp. 473-484, 1992.
 [3] G. Wei, J. Kim, D. Liu, S. Sidiropoulos, and M. Horowitz, "A variable-frequency parallel I/O interface with adaptive power-supply regulation," *IEEE Journal of Solid-State Circuits*, Vol. 35, No. 11, pp. 1600-1610, Nov. 2000.
 [4] Y. Shin, K. Choi, and T. Sakurai, "Power Optimization of real-Time Embedded Systems on Variable Speed Processors," in *Proceedings of the International Conference on Computer-aided Design*, pp. 365-368, Nov. 2000.

- [5] L. Shang, L.-S. Peh, and N. K. Jha, "Dynamic voltage scaling with links for power optimization of interconnection networks," in Proceedings High-performance Computer Architecture Symposium, pp. 91-102, 2003.
- [6] M. PEDRAM AND J. M. RABAEY, *Power Aware Design Methodologies*, Kluwer Academic, 2002.
- [7] J. Luo, N. K. Jha, L.-S. Peh, "Simultaneous Dynamic Voltage Scaling of Processors and Communication Links in Real-Time Distributed Embedded Systems," IEEE Transactions of Very Large Scale Integration (VLSI) Systems, Vol. 15, No. 4, pp 427-437, Apr. 2007.
- [8] Axel Jantsch, Hannu Tenhunen, *Networks on Chip*, Kluwer Academic, ch. 3, 2003
- [9] T. Burd and R. Brodersen, "Energy Efficient CMOS Microprocessor Design," in Proceedings of Hawaii International Conference on System Sciences, 1995.
- [10] J. Kim and M. Horowitz, "Adaptive supply serial links with sub-1 V operation and per-pin clock recovery," IEEE Journal of Solid-State Circuits, Vol. 37, pp. 1403-1413, Nov. 2002.
- [11] A. Chilambuchelvan, S. Saravanan and J.Raja Paul Perinbam, "A Simulation Software Development for Performance Analysis of DVS Algorithm for Low Power Embedded System," IEEE Transactions of Very Large Scale Integration (VLSI) Systems, Vol. 14, No. 2, pp.173-182, Feb. 2006.
- [12] L. Kiss and A. R. Várkonyi-Kóczy, "A Hybrid Autonomous Robot Navigation Method Based on Artificial Intelligence and Soft Computing Techniques," in Proceedings of IFAC International Conference on Intelligent Control Systems and Signal Processing, Apr. 2003.

저 자 소 개



노 경 우 (盧 暉 佑)

1978년 10월 23일생. 2006년 숭실대 컴퓨터학부 졸업. 2008년 동 대학원 컴퓨터학과 석사과정 졸업.

Tel : 02-813-0682

E-mail : nkw@ic.ssu.ac.kr



박 창 우 (朴 彰 雨)

1982년 7월 2일생. 2008년 숭실대 컴퓨터학부 졸업. 현재 동 대학원 컴퓨터학과 석사과정



김 석 운 (金 錫 潤)

1958년 8월 12일생. 1980년 서울대 공대 전기공학과 학사. 1990년 University of Texas at Austin 전기, 컴퓨터학과 석사과정 졸업. 1993년 University of Texas at Austin 전기, 컴퓨터학과 박사과정 졸업.

1982년~1987년 한국전자통신연구소 연구원.

1993년~1995년 Motorola Inc., Senior Staff Engineer. 1995년~현재 숭실대 컴퓨터학부 교수.