

퍼지 추론을 이용한 하드디스크드라이브의 유휴시간 최적화

論 文

57-3-22

Fuzzy Inference for Idle Time Optimization of Hard Disk Drive

全 鎮 完* · 李 知 炯† · 金 圭 澤**

(Jin-Wan Jun · Jee-Hyong Lee · Gyu-Taek Kim)

Abstract - Generally, HDDs are widely used as data storage device in office, home and mobile machineries. So, it is used for various applications or tasks, such as file copy, file download, music and movie play etc., in various environment. In spite of this kind of varieties in tasks and environment in which HDDs perform, most commercial HDDs hardly control its operations adaptively to these varieties. Thus, it is preferred to optimize the performance and energy consumption of HDDs according to the task and/or the environment. So, this paper proposes a new fuzzy inference algorithm which adaptively controls HDDs operations and may also easily be implemented as the firmware of HDDs and run in the restricted environment such as embedded systems.

Key Words : Fuzzy inference, Idle time optimization, Hard disk drive, Energy consumption

1. 서 론

노트북 컴퓨터는 LCD, CPU, ODD, RAM, HDD, network card 등 여러 부품들로 구성되며, 이러한 부품들은 각각 에너지를 소비하며 동작한다. 특히 HDD 경우에는 전체 시스템에서 소비하는 에너지의 약 18%를 사용하고 있다[1]. 그리고, HDD가 점차 소형화됨에 따라 외장형 저장장치, PMP, MP3 player 등 다양한 환경에서 사용되고 있다. 이로 인하여 HDD의 에너지소비를 최적화하게 되면 모바일 기기의 사용시간 개선이 가능해짐에 따라 모바일 환경에서의 에너지소비 최적화는 중요한 문제로 대두되고 있다. 하지만 에너지소비만을 최적화하면 성능이 떨어지며, 반대로 성능만을 최적화하면 에너지소비가 높아지는 문제점이 발생한다. 그래서, HDD가 현재 사용되는 환경에 따라서 적응적으로 두 항목을 조정해 주는 것이 중요하다[2]. 만약 HDD가 음악 감상 또는 문서작업처럼 전송되는 데이터가 적은 환경에서 사용되면 성능보다는 에너지소비가 중요한 요소가 된다. 그러나, 파일복사 또는 영화감상처럼 전송되는 데이터가 많은 환경에서 사용되면 성능이 중요한 요소가 되는 것이다. 이처럼 HDD가 사용되는 용도에 따라 중요한 항목이 달라진다.

하지만, 현재까지의 연구는 HDD의 에너지소비 최적화를 위한 방법에만 국한되어 있었고, 이렇게 제안된 방법마저 임베디드 시스템에 적용하기에는 너무 복잡하게 설계되었다. 일례로, 에너지 소비를 최적화하기 위하여 필요한 데이터를

미리 읽어서 HDD로의 접근을 최소화하는 방법이 제안되었다[3]. 하지만 HDD 자체에서는 다음에 읽을 데이터를 정확하게 예측하기가 어렵기 때문에 운영체제(OS) 같은 외부 응용프로그램의 도움 없이는 HDD 펌웨어 단독으로 구현하기는 어렵다. 다른 연구에서는 최근 접근지연 패턴을 분석하여 스피ن모터를 정지시키는 방법을 제안하였다[4][5]. 하지만, HDD가 작업 중에 스피ن모터를 정지할 만큼 충분한 지연이 발생하지 않기 때문에 실제 구현에 한계가 있다. 또한, 적응적 학습트리[6]를 사용하거나 마코프모델[1][7]을 이용한 연구가 진행되었지만, 이 방법들은 계산시 많은 자원이 소비되며 실시간 계산이 어렵기 때문에 제한된 자원을 가진 임베디드 시스템에 적용하기에는 적당하지 않다. 이처럼 기존 연구에서는 HDD가 사용되는 환경에 적응적으로 동작 가능하도록 제어하는 방법에 대해서 연구된 내용이 적고, OS의 도움을 받아야만 적용 가능한 방법만을 제시하였다. 본 논문에서는 외부 응용프로그램의 도움 없이 HDD 자체적으로 HDD가 사용되는 환경을 파악하여, 사용되는 환경에 따라 적응적으로 성능에 최적화되도록 HDD 유휴시간(비동작 상태)을 조정하거나, 에너지소비에 최적화되도록 유휴시간을 조정하도록 하는 퍼지추론 방법을 제시하고자 한다. 또한 이렇게 제시된 방법은 제한된 자원을 가진 임베디드 시스템에서 쉽게 구현 가능하며 퍼지추론에 의한 추가 연산 비용을 최소화 하도록 구성하였다.

2. HDD 및 퍼지 추론

2.1 HDD 의 동작

HDD 동작은 크게 두 가지 상태(state)로 구성 된다. 그것은 요구된 읽기/쓰기 명령을 수행하는 동작상태(ACTIVE)와

* 正 會 員 : 성균관대 반도체디스플레이공학 석사과정

† 교신저자, 正 會 員 : 성균관대 정보통신공학부 교수

E-mail : jhlee@ece.skku.ac.kr

** 非 會 員 : 삼성전자 스토리지 사업부 수석연구원

接受日字 : 2007年 12月 20日

最終完了 : 2008年 1月 21日

요구된 명령 완료 후 진입하는 비동작상태(IDLE)로 나뉜다. 여기서, 비동작상태는 다시 여러 단계의 비동작단계(idle phase)로 세분화할 수 있다[8][9][10]. 본 논문에서는 HDD 설계의 단순화를 위하여 그림 1처럼 idle1, idle2, idle3의 3단계 비동작단계 모델을 사용하였다. HDD는 명령이 요구되면 동작 상태에서 작업을 수행하고, 이 작업이 완료되면 바로 idle1 단계로 천이된다. 그리고, idle2는 idle1 단계 진입 후 새로운 명령 없이 일정시간이 지나면 천이되고, idle3로는 idle2 단계 진입 후 새로운 명령 없이 일정시간이 지나면 천이된다. 즉, idle2와 idle3 단계는 각각 자신의 타이머를 가지고 있고 이 값이 만료되면 HDD는 해당하는 단계로 천이된다. 그리고 새로운 명령이 들어오면 HDD는 동작 상태로 돌아가게 되며 idle2와 idle3 타이머 값은 초기화되어 다시 타이머를 구동하게 된다. 즉, 명령완료 후 HDD는 바로 idle1 단계로 천이되고 새로운 명령이 오기를 기다린다. 만약 새로운 명령이 요구되지 않고 idle2 타이머가 만료되면 HDD는 idle2단계로 천이하게 된다. 또한, 그림 1처럼 idle2에서 idle3로의 천이 역시 비슷한 방식으로 이루어진다. HDD에서는 이처럼 상태 천이가 항상 반복적으로 일어난다.

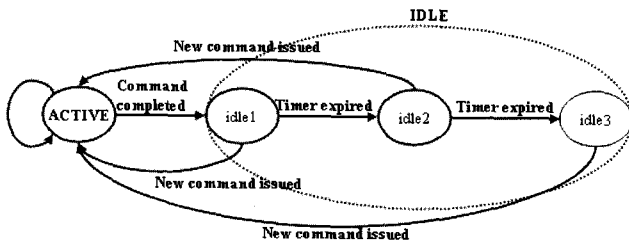


그림 1 HDD 동작모드
Fig. 1 HDD operation mode

각각의 비동작단계 차이점은 동작되지 않는 전기소자의 개수와 기구의 움직임에 따라 나뉘게 된다. 대부분의 전기소자가 동작되지 않고 기구가 안전영역으로 회피 되어있는 idle3 단계에서는 에너지소비가 최소화된다. 그러나, idle2 단계에서 idle3로의 천이 또는 idle3 단계에서 동작 상태로의 복귀는 많은 시간이 소요된다. 이와는 다르게 대부분의 전기소자 및 기구가 움직이는 idle1 단계에서는 새로운 명령에 대해서 빠르게 응답이 가능하기 때문에 성능이 최적화된다. 만약 idle2와 idle3이 긴 타이머를 가지고 있다면, HDD는 idle1 또는 idle2 단계에 있을 확률이 높기 때문에 새로운 명령에 대해서 빠른 응답을 보장할 것이다. 그러므로 이를 통하여 좋은 성능을 가질 수 있고 이것은 파일복사와 영화감상에 유리하다. 반면 HDD는 에너지소비를 많이 한다. 반대로 짧은 타이머를 가진 HDD는 idle2 또는 idle3 단계로 천이가 자주 일어나기 때문에 에너지소비를 적게 한다. 반면 새로운 명령에 대해서는 동작 상태로 복귀하는데 많은 시간을 소비하기 때문에 응답성능이 저하될 수 있다. 이것은 HDD가 음악 감상 목적으로 사용될 때 좋은 동작방식이다. 보통, 동작 상태에서 idle1 단계로 천이하는 시간은 idle1에서 idle2로 천이하는 시간보다 짧으며, idle2에서 idle3 단계로 천이하는 시간이 가장 길다. 이와 비슷하게 idle1에서 동

작 상태로 응답시간이 가장 짧으며, idle3에서 동작 상태로 응답시간이 가장 길다. 그림 2를 보면 각 idle 단계에서 응답시간과 에너지 소비와의 관계를 볼 수 있는데, 두 항목은 반비례의 관계를 나타내기 때문에 적용적으로 동작되는 HDD 설계를 위해서는 각 idle 단계 타이머 값을 조절하는 방법이 가장 중요한 설계요소가 된다.

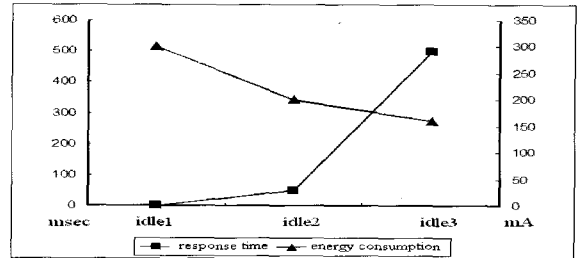


그림 2 에너지소비와 응답시간의 관계
Fig. 2 Relationship between energy consumption and response time

2.2 퍼지 추론

설계된 퍼지추론은 두개의 입력 변수를 사용한다. 첫째 입력 변수는 요구된 데이터 비율을 나타내는 RRD(Ratio of the requested data)이며, 둘째 변수는 비동작단계 비율을 나타내는 RIP(Ratio of idle phase)이다. 이를 이용하여 새로운 비동작단계 타이머 값을 얻게 된다. 본 논문에서 사용된 퍼지추론에 필요한 입력 변수 데이터는 5분간 수집되며, 이렇게 수집된 데이터를 이용하여 퍼지추론을 수행하였다. 여기서, 퍼지추론 계산주기는 계산에 소요되는 시간과 계산주기와의 비율을 이용하여 결정 가능하다. 만약, 퍼지추론의 계산에 100ms가 소비 된다면, 5분 간격으로 퍼지추론을 수행하는 HDD의 경우 0.033%의 추가 비용이 필요하게 된다. 이러한 추가 비용은 전체 성능에 큰 영향을 주지 않으므로 사용 가능한 값이 된다.

2.2.1 퍼지 규칙

입력 변수로 사용되는 RRD은 일정한 데이터 수집시간 동안 요구된 데이터 블록 개수와 설계최대 데이터 블록 개수의 비율로 나타내며, 식(1)에 의해서 계산된다.

$$RRD = \frac{DataBlock_{total}}{DataBlock_{max}} \quad (1)$$

$DataBlock_{total}$ = 데이터 수집시간 동안 요구된 데이터 블록 개수

$DataBlock_{max}$ = 설계최대 데이터 블록 개수

이 값은 HDD에 얼마나 많은 데이터가 요구되는가에 대한 정보를 나타낸다. 만약 RRD가 0에 근접하면 요구되는 데이터가 매우 적다는 의미가 된다. 일반적으로 소속도 합

수를 결정할 때 기준으로 삼는 것은 HDD가 오피스작업 용도로 사용될 때의 데이터를 이용한다. RRD의 경우는 그림 3에서 보이는 dMS가 오피스 작업할 때 나타나는 소속도 함수이며, 이보다 적은 데이터가 요구되는 음악 감상 같은 경우에는 dMS의 좌측 부분에 소속도 함수를 표현하고, 많은 데이터가 요구되는 파일복사의 경우에는 dMS의 우측 부분에 소속도 함수를 표현한다. 이를 이용하여 RRD에 대한 소속도 함수를 그림 3과 같이 나타내었다.

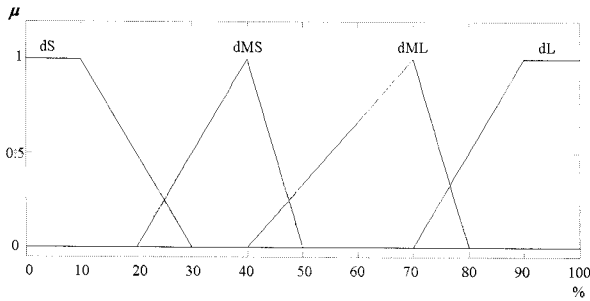


그림 3 요구된 데이터 비율(RRD) 소속도 함수
Fig. 3 Membership function for RRD

이제, 다른 입력 변수인 RIP를 알아보도록 하겠다. 만약 HDD가 일정시간 동안 작동될 때 HDD 내부의 컨트롤러는 idle2와 idle3 단계에서 얼마 동안 머물렀는지 간단히 계산 가능하다. 이 값에 식(2), 식(3)을 이용하면 각 idle 단계에서 머무른 시간에 대한 비율을 계산할 수 있다. 그리고 이 값을 이용하여 입력 변수로 사용할 RIP값은 식(4)을 이용하여 계산한다.

$$idle2_{prop} = \frac{idle2_{total}}{Runtime_{total}} \quad (2)$$

$$idle3_{prop} = \frac{idle3_{total}}{Runtime_{total}} \quad (3)$$

$$RIP = idle2_{prop} + idle3_{prop} \quad (4)$$

$Runtime_{total}$ = 정의된 데이터 수집시간
 $idle2_{total}$ = idle2 단계에서 머문 전체시간
 $idle3_{total}$ = idle3 단계에서 머문 전체시간

이 값은 얼마나 자주 명령이 요구 되었는지에 대한 정보를 나타낸다. 만약 RIP값이 100에 가깝다면, 일정시간 동안 명령이 거의 요구되지 않았다는 의미이다. 이러한 소속도 함수를 구성하기 위해서는 HDD가 사용되는 환경에서 각각의 비동작단계에서 머무른 시간에 대한 분석이 필요하다. 일례로, 파일복사의 경우는 비동작단계 보다는 동작단계가 90%이상 차지하며, 음악 감상의 경우는 비동작단계가 98% 이상 차지한다. 이렇게 각 경우에 대한 대표적인 데이터를 이용하여 RIP의 소속도 함수를 결정하고, 이렇게 결정된 소속도 함수를 그림 4와 같이 나타내었다.

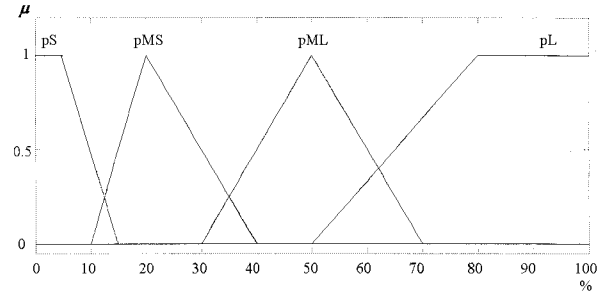


그림 4 비동작단계 비율(RIP) 소속도 함수
Fig. 4 Membership function for RIP

식(4)은 idle2 단계를 위한 것인데, 만약 idle3 단계에 대한 타이머를 추론하고자 하면 식(5)을 식(4) 대신 사용한다.

$$RIP = idle3_{prop} \quad (5)$$

결론부에서는 새로운 비동작단계 타이머 값 IPT(Idle phase timer)을 얻게 된다. 결론부에서 사용되는 소속도 함수는 그림 5와 같다. 만약, 식(4)이 입력변수로 사용되었다면 결론부에서는 idle2에 대한 새로운 타이머 값을 얻게 되며, 식(5)을 사용하였다면 idle3에 대한 새로운 타이머 값을 얻는다. 이 소속도 함수 역시 HDD가 오피스작업에 사용될 때를 기준으로 삼는다. 여기서는 오피스작업의 경우 그림 5에 나타나는 fS와 fM 중간 값으로 추론되도록 구성 하였다. 이 값은 에너지소비와 응답성능이 평형 상태에 있는 실험적인 값을 사용하게 된다. 그리고, 적은 데이터가 요구되는 음악 감상은 이 값보다 작은 값으로 추론되도록 소속도 함수를 생성하고, 파일복사의 경우는 이 값보다 큰 값으로 추론되도록 하였다. 단, 데이터가 적게 요구되어도 자주 요구되는 경우는 응답성능을 보장하기 하기 위하여 비동작단계 타이머 값이 커지도록 설계를 한다. 표 1에서 IPT에 대한 퍼지규칙을 보였는데, 여기서 첫째 규칙이 의미하는 것은 요구되는 데이터 블록의 양이 적고 비동작단계 비율이 낮다면 중간 정도의 타이머 값을 얻게 되는 것이다. 이 규칙은 읽기/쓰기가 적지만 자주 들어올 때 적용이 가능하다.

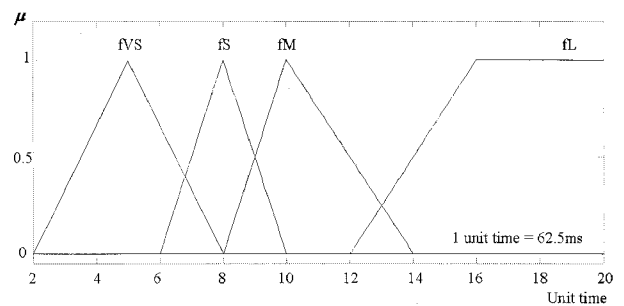


그림 5 결론부 소속도 함수
Fig. 5 Membership function for IPT

표 1 결론부의 규칙

Table 1 If-then rules

Rule 1	If(RRD is dS) and (RIP is pS) then IPT is fM
Rule 2	If(RRD is dS) and (RIP is pMS) then IPT is fS
Rule 3	If(RRD is dS) and (RIP is pML) then IPT is fS
Rule 4	If(RRD is dS) and (RIP is pL) then IPT is fVS
Rule 5	If(RRD is dMS) and (RIP is pS) then IPT is fL
Rule 6	If(RRD is dMS) and (RIP is pMS) then IPT is fM
Rule 7	If(RRD is dMS) and (RIP is pML) then IPT is fS
Rule 8	If(RRD is dMS) and (RIP is pL) then IPT is fVS
Rule 9	If(RRD is dML) and (RIP is pS) then IPT is fL
Rule 10	If(RRD is dML) and (RIP is pMS) then IPT is fM
Rule 11	If(RRD is dML) and (RIP is pML) then IPT is fM
Rule 12	If(RRD is dML) and (RIP is pL) then IPT is fS
Rule 13	If(RRD is dL) and (RIP is pS) then IPT is fL
Rule 14	If(RRD is dL) and (RIP is pMS) then IPT is fL
Rule 15	If(RRD is dL) and (RIP is pML) then IPT is fM
Rule 16	If(RRD is dL) and (RIP is pL) then IPT is fM

2.2.2 신규 명령구간 회피

만약, HDD가 동작 상태에서 idle1 단계 또는 idle2 단계에서 idle3 단계로 천이 중 이라면, 새롭게 요청된 명령은 천이 동작이 완료될 때 까지는 응답을 못하게 된다. 즉, 천이되는 중간에 명령이 자주 요구되면 응답 성능은 현저히 떨어지게 된다. 또한 천이 동작에는 일정한 에너지가 소비되므로 다음단계로 이동 후 소비된 에너지를 상쇄할 만큼 충분히 머무르지 못하면 에너지소비 측면에도 좋지 않다. 그러므로 우리는 명령이 자주 요구되는 구간에 대한 고려가 필요하다. 다음 명령이 언제 요구되는지 예측하기 위해서, 여기서는 매번 명령이 요구될 때마다 이전 명령과 현재 명령과의 명령이 간격을 기록한다. 이때, 이러한 기록 동작은 퍼지입력 변수 데이터가 수집되는 기간과 동일한 시간 동안 수행되며, 퍼지추론이 수행된 후에는 새롭게 명령어 간격을 기록하게 된다. 이렇게 기록된 데이터를 가지고 그림 6(a)과 같은 히스토그램을 만든다. 이 히스토그램에서 x축은 명령어 간격의 길이를 나타내고, y축은 해당하는 명령어 간격에서 요구된 명령어 개수를 나타낸다.

작성된 히스토그램을 이용하여 명령이 요구된 구간에 대하여 다음의 방법으로 퍼지화를 수행한다. 우선, 빈도수가 가장 큰 계급을 선택하고 이 값을 현재최대계급값으로 설정한다. 이 값을 $\max(\mu_1(x))=1$ 가지며 삼각형 퍼지 함수의 꼭짓점으로 설정된다. 그리고 $\mu_1(x)=0$ 해당하는 우측 값은 식(6)을 이용하여 현재최대계급 보다 일정 비율만큼 작아지는 계급을 선택한다. 그리고 좌측 값은 우측 값 설정 방법과 동일한 과정을 거치면서 설정한다. 이런 방식을 사용하면 그림 6(b) 같은 신규 명령에 대한 하나의 퍼지함수가 만들어진다. 퍼지함수가 만들어진 후 좌/우측 경계 값 사이에 속하는 계급은 모두 0으로 설정한다. 이후 동일한 방법을 사용하여 새로운 현재최대계급 값을 설정 후 다른 퍼지함수를 만들 수 있다. 그림 6(c)에 두 번째로 만들어진 퍼지함수를 나타내었다. 이때, 퍼지 함수의 최대값은 식(7)을 이용하여 조정한다. 이렇게 하여 최종적으로 퍼지화된 결과를 그림 6(d)에 보이고 있다.

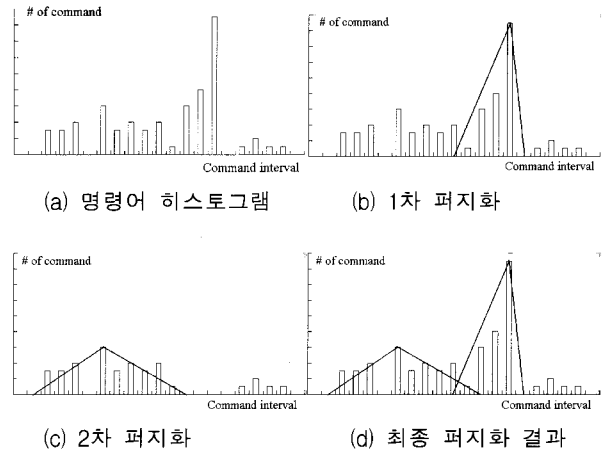


그림 6 신규명령구간 소속도 함수

Fig. 6 Membership function of new command

$$Boundary = \text{현재최대계급값}_n \times 0.25 \quad (6)$$

$$\max(\mu_n(x)) = \frac{\text{현재최대계급값}_n}{\text{현재최대계급값}_1} \quad (7)$$

이렇게 퍼지화된 소속도 함수는 다음 명령이 언제 요구되는지에 대한 가능성을 나타낸다. 그러므로, 이렇게 만들어진 퍼지함수 영역에서는 비동작단계 천이가 일어나는 것을 피해야 한다. 그러므로, 이를 위하여 우리는 식(8)을 이용하여 결론부의 퍼지결과와 연산을 수행 하도록 하였다.

$$C = A - B = A \cap B^C \quad (8)$$

퍼지집합 A는 결론부의 퍼지규칙에 의해서 나타나고, 퍼지집합 B는 그림 6(d)에서 보여준 신규 명령구간에 대한 퍼지화 결과이다. 퍼지집합 B의 여집합은 그림 7과 같이 퍼지 연산을 이용하여 구할 수 있다. 이 소속도 함수는 비동작단계로 천이 가능한 영역을 표현하게 된다.

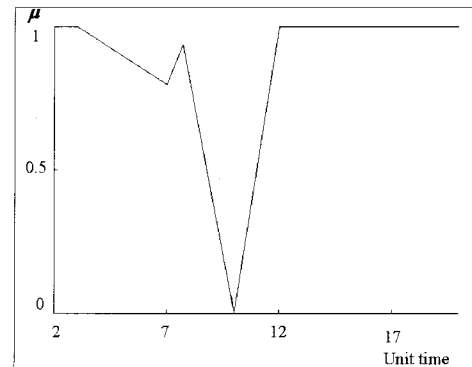


그림 7 신규명령구간 소속도 함수의 여집합

Fig. 7 Complementary set of fuzzified histogram

예를 들어 명령이 요구된 후 100ms후에 다시 명령이 요구될 가능성이 높다면 우리는 그 시간 주위에서는 비동작단계 천이가 일어나는 것을 피해야 할 것이다.

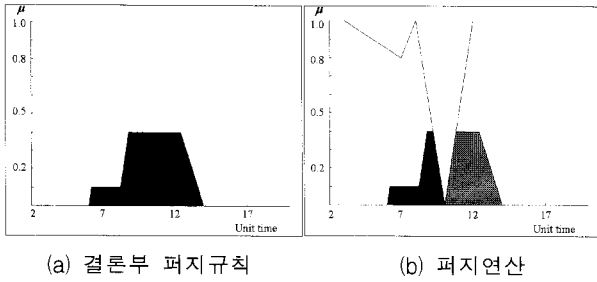


그림 8 비 퍼지화
Fig. 8 Defuzzification

그림 8(a)은 퍼지집합 A를 그림 8(b)은 퍼지집합 C를 나타내는 예이다. 만약 그림 8(b)처럼 완전히 분리되는 영역으로 나뉘었다면 비퍼지화는 분리된 영역 중 면적이 가장 넓은 영역만 수행하여 그 결과를 이용한다[11][12].

3. 실험

우리는 비동작단계 타이머를 최적화하는 퍼지 추론을 제안하였다. 기존에 HDD가 동작 상태에서 비동작상태로 천이하는 방법은 고정된 타이머를 사용하여 이 시간 동안 새로운 명령이 없으면 자동으로 비동작상태로 천이하는 방식(FIXED) 사용하였다. 본 실험에서는 이러한 기존방법과 더불어 명령어를 고려하지 않은 퍼지 추론(FUZZY/a)과 명령어를 고려한 퍼지 추론(FUZZY)등 3가지 방법을 비교실험하고자 한다. 이때 퍼지 추론은 idle2 단계 천이와 idle3 단계 천이에 대하여 모두 적용하여 실험을 수행 하였다. 그리고, 각각의 방법에 대해서 파일복사, 영화감상, 음악 감상과 문서 작업 동시수행, 오피스작업, 음악 감상 등 다섯 가지 용도로 HDD가 사용될 때 응답성능 및 에너지소비가 어떻게 변하는지 보이고자 한다. 본 실험에서는 HM250JI SAMSUNG 2.5"250GB HDD 모델을 테스트에 사용하였고 표 2에 자세한 성능을 표시 하였다[13].

표 2 HM250JI 성능표

Table 2 Specification of HM250JI

	응답시간	에너지소비
Active	N/A	400mA
idle1	<1ms	300mA
idle2	<50ms	200mA
idle3	<500ms	160mA

다섯 가지 사용용도 시나리오는 윈도우 상에서 실제 관련 응용프로그램을 수행하면서 읽기/쓰기 명령이 언제 요구되는지, 명령과 다음 명령 사이의 명령어 시간 간격을 명령어 분석도구를 이용하여 30분 동안 명령어를 분석 하였다[14]. 이렇게 분석된 데이터를 이용하여 실행 가능한 스크립트를 제작하였기 때문에, 수행되는 테스트는 실제 사용 환경을 잘 반영하고 있다.

3.1 에너지 소비 평가

에너지소비는 각각의 시나리오를 수행하면서 소비된 평균

전류로 표현 하였다. 그림 9 및 표 3에 각각의 시나리오에서 에너지소비 결과를 보여준다.

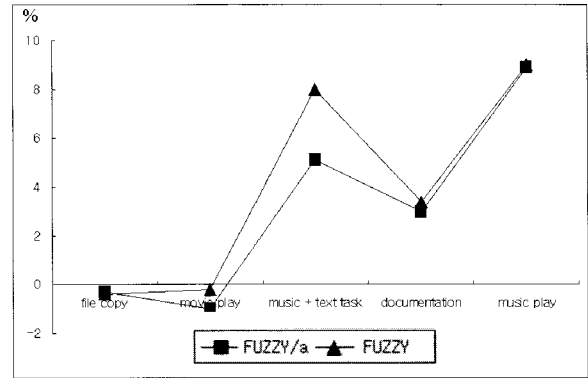


그림 9 에너지 소비 개선비율(%)

Fig. 9 Improvement ratio of energy consumption

표 3 HDD 사용 용도별 에너지 소비

Table 3 Energy consumption comparison(mA)

Task	FIXED	FUZZY/a	FUZZY
파일복사	407.8	409.1	409.5
개선율(%)	N/A	-0.3	-0.4
영화감상	278.1	280.8	278.5
개선율(%)	N/A	-1.0	-0.2
음악+문서	240.2	228.5	221.8
개선율(%)	N/A	5.1	8.0
오피스작업	237.6	230.8	230.1
개선율(%)	N/A	3.0	3.4
음악 감상	227.1	208.2	208.1
개선율(%)	N/A	8.9	9.0

음악 감상과 문서작업 동시수행, 오피스작업 그리고 음악 감상처럼 적은 데이터가 요구되는 작업에서는 에너지소비가 기존보다 좋아진 것을 볼 수 있다. 이는 적은 데이터가 요구되는 작업이기 때문에 응답성능 보다는 에너지소비가 좋아지는 상태로 전환되었기 때문이다. 즉, 각각의 비동작단계 설정 타이머 값이 퍼지추론에 의해서 기존보다 작은 값으로 설정된 것이다. 반면 많은 데이터가 요구되는 작업에 해당하는 파일복사 및 영화감상에서는 기존과 유사하거나 오차 범위 내에서 약간 높은 에너지가 소비된 것을 볼 수 있다. 이러한 결과는 많은 데이터가 요구되는 상황에서는 응답성능을 우선하는 상태로 전환되기 때문이다. 여기서 특징적으로 보아야 할 것은 음악 감상과 문서작업 동시수행 시나리오이다. 이 시나리오 이외 다른 것들은 명령어를 고려한 퍼지추론과 고려하지 않은 퍼지추론의 에너지소비 결과에 큰 차이가 없다. 하지만 이 시나리오는 명령어를 고려하지 않은 퍼지추론의 경우 5.1%의 개선이 있으나 명령어를 고려한 퍼지 추론은 8.0%의 개선을 보여주어 더 좋은 에너지소비를 보여준다. 이렇게 차이가 발생하는 원인은 명령어를 고려하지 않은 퍼지추론의 경우에는 비동작단계로 천이할 때 새로운 명령어가 많이 요구되었기 때문이며, 명령어를 고려한 경우에는 새로운 명령어가 자주 들어오는 지점을 피하여 비동작단계로 천이가 일어나기 때문에 발생하는 것이다.

3.2 응답성능 평가

본 논문에서는 각 명령어에 필요한 평균 지연시간을 응답성능으로 정의한다. 그리고, 이 응답성능에 대한 개선율은 식(9)을 이용하여 계산한다.

$$Ratio = \frac{Org_{time} - Fuzzy_{time}}{Org_{time}} \times ACTIVE_{ratio} \quad (9)$$

계산시 HDD가 실제 동작된 비율($ACTIVE_{ratio}$)을 고려해주는 이유는 HDD가 실제 사용되는 시간 이외의 비동작 시간은 응답성능에 전혀 영향을 미치지 않으므로, 이 부분을 제외하고 계산해야 전체 시스템의 평균 응답성능이 나오기 때문이다. 식(9)에서 Org_{time} 은 기존 방식이 사용 되었을 때 평균 응답성능을 나타내며, $Fuzzy_{time}$ 은 퍼지추론이 사용 되었을 때 값을 나타낸다. 그림 10 및 표 4는 응답성능에 대한 비교결과를 보여준다.

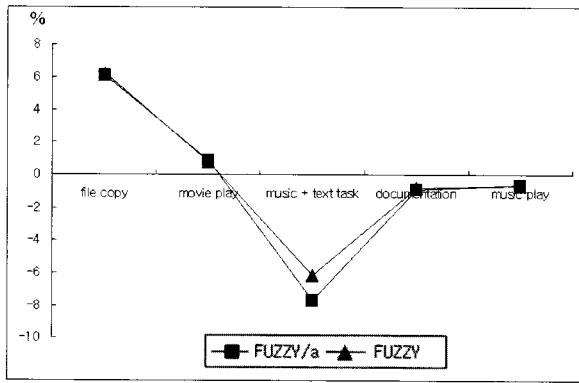


그림 10 응답 성능 개선비율(%)
Fig. 10 Improvement ratio of the response performance

표 4 HDD 사용 용도별 평균 응답 성능

Table 4 Response performance comparison(ms/cmd)

	FIXED	FUZZY/a	FUZZY	ACTIVE
파일복사	3.35	3.03	3.03	90.8
개선율(%)	N/A	6.1	6.1	
영화감상	7.3	4.4	4.8	2.4
개선율(%)	N/A	0.9	0.8	
음악+문서	15.5	71.6	61.2	2.1
개선율(%)	N/A	-7.7	-6.2	
오피스작업	32.6	37.3	36.7	7.0
개선율(%)	N/A	-0.9	-0.8	
음악 감상	62.3	85.5	89.8	1.9
개선율(%)	N/A	-0.6	-0.7	

많은 데이터가 요구되는 작업에 해당하는 파일복사 및 영화감상은 기존 방법보다 응답성능이 개선된 것을 볼 수 있다. 이것은 비동작단계 변경 타이머 값이 퍼지추론에 의해서 기존보다 큰 값으로 설정되었기 때문이다. 음악 감상과 문서작업 동시수행 그리고 음악 감상처럼 적은 데이터가 요구되는 작업의 경우에는 응답성능이 기존보다 떨어지는 결

과를 보인다. 이것은 응답성능 보다는 에너지 소비를 좋게 하는 상태로 변경되었기 때문이다. 음악 감상과 문서작업 동시수행 시나리오는 명령어를 고려하지 않은 퍼지 추론의 경우 -7.7% 응답 성능이 떨어지나, 명령어를 고려한 경우는 -6.2% 정도만 떨어지는 것으로 나타난다. 이는 명령어를 고려하면 조금 더 우수한 결과를 얻게 됨을 알 수 있다. 그러나, 여기서 발생하는 성능 저하는 실제 모바일 기기를 사용하는 사용자가 체감할 수 있는 정도는 아니다. 예를 들어 3분의 플레이 시간을 가진 10MB MP3 파일이 재생된다고 하면, 음악 감상용 프로그램은 평균 60KB/s 정도의 데이터가 필요하다. 그러므로, 모바일 기기가 100KB/s의 버퍼링 기능만 가지고 있어도 재생이 끊어지는 문제는 발생하지 않는다. 또한, HDD는 하나의 명령어에 대해서 100ms의 평균 지연을 가지고 있다고 하여도 최대 1MB/s 정도의 성능을 낼 수 있기 때문에 HDD로 요구되는 데이터를 충분히 처리 가능하다. 이처럼 적은 데이터가 요구되는 상황에서의 성능 저하는 문제가 발생하지 않는 수준이다.

그림 11에 나타난 idle2의 변화를 이용하여 실험 결과를 정리해 보면 다음과 같다. 제안된 퍼지추론은 음악 감상에서 9%, 음악 감상과 문서작업 동시수행은 8%의 에너지소비 향상 결과를 보여주었다. 이것은 적은 데이터가 요구되는 작업의 경우에는 그림 11에서 보여지는 것처럼 비동작단계 변경 타이머 값이 시간이 지나면서 최초 설정된 값보다 작은 값으로 변경되기 때문이다. 그리고, 많은 데이터가 요구되는 파일복사의 경우에는 6.1%의 성능향상을 보여주었다. 이는 그림 11에서 보여지는 것처럼 비동작단계 변경 타이머 값이 최초 설정된 값보다 큰 값으로 변경되기 때문이다. 또한, 명령어를 고려하지 않을 때 보다 고려했을 때 음악 감상과 문서작업 동시수행 시나리오에서 3%의 추가 에너지소비 향상과 1.5%의 응답성능 향상 결과를 보여주는데, 이는 명령어를 고려하게 되면 그림 11처럼 비동작단계 변경 타이머 값이 명령어가 들어오는 구간을 피하여 설정되기 때문이다. 이처럼 퍼지 추론에 의하여 HDD가 사용되는 환경에 적응적으로 동작되는 모습을 볼 수 있다.

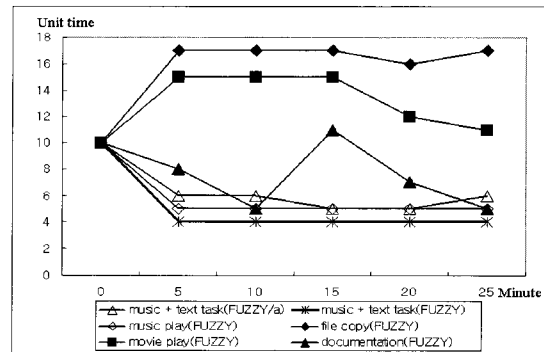


그림 11 시나리오에 따른 idle2 timer 추이
Fig. 11 Idle2 timer trend per scenarios

4. 결론

노트북 또는 많은 모바일 기기로 인하여 HDD가 사용되는 용도는 점점 다양해지고 있다. 이를 위하여, 본 논문에서

는 HDD가 사용되는 용도에 따라 적응적으로 에너지소비 및 응답성능을 제어하는 방법을 제시 하였고 만족할 만한 결과를 보여주었다. 이러한 결과는 다양한 사용자 환경에서 HDD가 항상 최적의 성능을 발휘 하도록 해준다. 또한, 제안된 방법은 HDD처럼 제한된 자원을 가진 임베디드 시스템에 쉽게 적용이 가능하다. 이로 인하여 HDD 펌웨어 변경만으로 사용되는 운영체제와 독립적으로 그 기능을 수행할 수 있다. 앞으로 신규명령어에 대한 소속도 함수를 만드는 방법으로 클러스터링 방법을 이용하고 각각의 비동작단계에서 동작 상태로 복구할 때 필요한 응답시간 까지 고려한 소속도 함수를 만들면 더욱 좋은 결과를 보이는 퍼지 추론을 구성 할 수 있다.

참 고 문 헌

[1] L. Benini, and A. Bogliolo, "Policy Optimization for Dynamic Power Management", IEEE Transactions on computer-aided design of integrated circuits and system, Vol.18, No.6, 1999, pp.813-833.

[2] P. Greenawalt, "Modeling Power management for Hard Disks", Proc. Int'l Workshop Modeling, Analysis, and Simulation for Computer and Telecomm. System, 1994, pp. 62-65

[3] A.E. Papathanasiou, and M.L. Scott, "Energy efficient prefetching and caching", Proc. USENIX Annu. Tech. Conf, 2004, pp. 255-268

[4] C.H. Hwang and A. Wu, "A Predictive System Shutdown Method for Energy Saving of Event-Driven Computation", Proceedings of the international conference on computer aided design, 1997, pp.28-32.

[5] F. Douglis, P. Krishnan, and B. Bershad, "Adaptive Disk Spin Down Policies for Mobile Computers", Computing Systems, Vol.8, 1995, pp.381-413.

[6] E.Y. Chung, L. Benini, and G.D. Micheli, "Dynamic power management using adaptive learning tree", Proc. Int. Conf. Comput.-Aided Des., 1999, pp.274-279.

[7] E.Y. Chung, L. Benini, A. Bogliolo, Y.H. Lu, and G.D. Micheli, "Dynamic power management for nonstationary service requests", IEEE Trans. Comput. Vol.51, No.11, 2002, pp.1345-1361.

[8] IBM, "Adaptive Power Management for Mobile Hard Drives", White paper, 1999.

[9] W.F.Heybruck, "Enhanced adaptive battery life extender (ABLE)", White paper, HGST, 2005.

[10] Q. Zhu, F.M. David, C.Devaraj, Z.Li, Y.Zhou, and P.Cao, "Reducing Energy Consumption of Disk Storage Using Power-Aware Cache Management", Proc. Int. Symp. High-Performance Comut. Architecture, 2004, pp118-129.

[11] G. Bojadziev, and M. Bojadziev, Fuzzy sets, Fuzzy logic and Applications, World Scientific, 1998

[12] J. Yen, and N. Pfluger, "A Fuzzy Logic Based Extension to Payton and Rosenblatt's Command fusion Method for Mobile Robot Navigation", IEEE Transactions on systems, man and cybernetics, Vol.25 No.6, 1995, pp. 971-978

[13] SAMSUNG, "http:// www.samsung.com/Products/HardDiskDrive/SpinpointMSeries"

[14] LeCroy, "http:// www.lecroy.com.com/tm/products/ProtocolAnalyzers", SAS/SATA Protocol Analyzers

저 자 소 개



전진완 (全鎭完)

1974년 3월 3일생. 1999년 성균관대학교 기계공학과 졸업. 2007년~현재 동 대학원 반도체디스플레이공학과 석사과정. 1999년~현재 삼성전자 스토리지 사업부 책임연구원
 Tel : 031-200-8209
 Fax : 031-200-3536
 E-mail : jwjeon@samsung.com



이지형 (李知炯)

1970년 3월 15일생. 1993년 한국과학기술원 전산학과 졸업. 1995년 동 대학원 전산학과 석사. 1999년 동대학원 전산학과 박사. 2000-2002년SRI(Stanford Research Institute), International Fellow, 2002년~현재 성균관대학교 정보통신공학부 교수
 Tel : 031-290-7154
 Fax : 031-290-7230
 E-mail : jhlee@ece.skku.ac.kr



김규택 (金圭澤)

1968년 8월 9일생. 1991년 서울대 컴퓨터 공학과 졸업. 1993년 한국 과학기술원 전산학과 석사. 1993~현재 삼성전자 스토리지 사업부 수석연구원
 Tel : 031-200-3621
 Fax : 031-200-3536
 E-mail : gt.kim@samsung.com