

적응적 큐브 분할을 이용한 하이퍼큐브 진단 알고리즘

(A Diagnosis Algorithm for Hypercube Multiprocessors using Adaptive Cube Partition Method)

최 문 옥 [†] 이 충 세 ^{**}

(Moon-Ok Choi) (Chung-Sei Rhee)

요 약 본 논문에서는 하이퍼큐브 병렬시스템을 진단하기 위한 알고리즘으로써 적응적 큐브 분할 방법을 적용한 진단 알고리즘을 제안한다. Feng[1] 등이 제안한 하이퍼큐브 진단 알고리즘은 이전의 연구결과에 비교하여 비교적 적은 진단 비용을 필요로 하는 반면 진단 알고리즘의 수행 중에 나타나는 큐브의 분할이 신드롬의 분석이 없이 정적인 방법을 사용하여 진단함으로써 불필요한 진단 오버헤드를 야기하였다. 본 논문에서는 이를 보완하기 위한 방법으로 신드롬 분석을 통하여 진단비용을 줄일 수 있는 분할을 선택하는 적응적 큐브 분할방법을 제안한다. 또한 시뮬레이션을 통하여 Feng이 제안한 방법에 비하여 효율적인 성능을 나타내는 것을 보인다.

Abstract In this paper, we propose a system-level diagnosis algorithm for hypercube multiprocessors using adaptive cube partition method. Feng[1] proposed a diagnosis algorithm for hypercube multiprocessors which gives a better performance compared to previous researches[2, 3]. But cube partitions in Feng's algorithm are performed without syndrome analysis. Therefore unnecessary overhead is made during cube partitions. In this paper, we propose an adaptive cube partition method which gives better partition through syndrome analysis and reduces diagnosis cost. We give a simulation result for comparisons. We have found that our algorithm shows better performance compared to Feng's method.

1. 서 론

병렬처리 시스템의 규모가 커짐에 따라 시스템 내에서 발생하는 결함의 빈도가 높아지고 있다. 또한 결함의 증가로 인하여 시스템의 다운과 수리 비용은 병렬처리 시스템의 성능을 저하시키는 큰 요인으로 부각되고 있다. 이에 따라서 병렬처리 시스템에서 발생할 수 있는 결함을 온라인으로 진단하여 시스템의 신뢰도와 가용성을 향상시키려는 많은 연구가 진행되어 왔으며, 결함의 진단에 필요한 비용을 최소화하려는 많은 연구가 필요

하다.

여러 가지 병렬컴퓨터 중에서 하이퍼큐브 구조의 병렬처리 시스템이 최근 들어 많이 사용되고 있다. 하이퍼큐브 구조는 정규적이며 계층적인 구조를 가지고 있는데, 이러한 특성은 효율적인 진단알고리즘 개발에 유용하게 이용될 수 있다. 따라서 본 논문에서는 하이퍼큐브 병렬처리 시스템에서 효율적인 진단을 수행할 수 있는 알고리즘을 제안한다.

본 논문에서는 잘 알려진 진단모델인 Preparata, Metz, 그리고 Chien에 의해 제안된 PMC 모델[2]을 사용한다. PMC 모델에서는 시스템 S 는 n 개의 장치로 나타내어지며, 각 장치는 시스템 내의 다른 몇 개의 장치에 의해 테스트를 받게된다. 시스템 S 를 구성하는 n 개의 장치와 테스트 관계는 방향그래프 $G=(V, E)$ 로 표현한다. 여기서 정점의 집합 V 는 시스템을 구성하는 장치(또는 노드)들을 나타내며, 간선의 집합 E 는 장치간의

[†] 비 회 원 : 충북대학교 컴퓨터학과
greenmax@algo.chungbuk.ac.kr

^{**} 종 신 회 원 : 충북대학교 컴퓨터학과 교수
csrhee@cbucc.chungbuk.ac.kr

논문접수 : 1999년 8월 9일

심사완료 : 2000년 2월 8일

테스트관계들을 나타낸다. 즉, i 에서 j 로의 간선 (i, j) 가 존재하면 정점 i 는 테스트를 수행하는 정점, 정점 j 는 테스트를 받는 정점을 나타낸다. 진단 알고리즘이 수행되고 나면 간선 (i, j) 는 테스트를 받는 정점 j 가 정상일 경우는 0, 결함이 발생했을 경우는 1의 값을 갖게 된다. 이와 같이 테스트가 수행되고 난 후 각 간선이 갖는 값들의 집합을 신드롬이라 정의한다. PMC 모델은 신뢰도가 높은 별도의 프로세서에 의하여 신드롬이 분석된다고 가정한다. 즉, 신드롬 분석을 담당하는 컴퓨터(Syndrome Analyzer)는 각 정점의 테스트가 수행된 후 얻어지는 신드롬을 분석하여 시스템 S 에 대한 진단을 하게 된다.

본 논문에서는 진단 비용을 계산하기 위하여 [1]에서 사용된 다음의 두 가지 값을 측정한다.

- (1) 진단 알고리즘의 수행시간(테스팅 라운드)
- (2) 진단하기 위하여 사용된 테스트 링크의 수

또한 발생한 결함은 영구적이며, 진단 알고리즘의 수행 중에는 새로운 결함이 발생되지 않는다는 가정과 함께 시스템의 진단이 온라인으로 수행됨을 가정한다. Feng은 시스템-레벨 하이퍼큐브 시스템을 적응적으로 진단하는 알고리즘인 HADA(Hypercube Adaptive Diagnosis Algorithm), IHADA(Improved HADA)[1]를 제안하였는데, 본 연구에서는 HADA, IHADA를 이용하여 하이퍼큐브 시스템을 진단할 경우 진단 비용을 줄일 수 있는 적응적 큐브 분할 방법을 제안한다. 제안한 방법은 HADA, IHADA와 비교하여 평균적으로 비교적 적은 테스트 단계와 테스트 링크를 사용하게 된다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 Feng에 의해 제안된 진단 알고리즘을 포함한 기존의 관련 연구에 대해서 알아본다. 3장에서는 본 논문에서 제안하는 적응적 큐브 분할 방법을 소개하고, 4장에서는 제안한 알고리즘에 대한 분석결과를 제시한다. 이어서 제안한 알고리즘에 대한 시뮬레이션 결과를 5장에서 살펴본 후, 6장에서는 결론을 제시한다.

2. 관련 연구

Nakajima는 적응적 진단 방법(Adaptive Diagnosis Scheme)을 이용한 시스템 진단알고리즘을 제안하였다 [4]. 적응적 방법에서는 시스템을 진단하기 위해서 수행되는 테스트가 다이나믹한 방법에 의하여 선택적으로 수행된다. 따라서 기존의 방법에 모든 테스트 링크를 사용하는 방법에 비하여 적은 수의 테스트 링크들만을 이용하여 진단을 수행할 수 있다는 장점을 가지고 있다.

그러나 Nakajima에 의한 적응적 진단 방법은 모든 노드들 간의 링크가 존재한다는 완전 연결된 시스템에서만 사용될 수 있다는 단점이 있다.

Vaidya와 Pradhan은 [4]에서 이용했던 노드간의 완전 연결에 대한 가정을 완화시킨 방법으로써 ASDA(Adaptive Safe Diagnosis Algorithm)[5]을 제안하였다. 그러나 ASDA를 적용할 경우 각 테스트 단계에서 불필요한 테스트가 수행될 수 있으며, 최악의 경우에는 모든 테스트가 수행되어야만 한다.

노드의 집합이 R 로 표현되는 링에서 테스트가 양방향(시계방향과 반시계방향)으로 수행될 경우, $1-FL-(|R|-2)-FD$ 의 성질이 있음을 알 수 있다[5]. 여기서 $t-FL-u-FD$ (t -fault-location- u -fault-detecting)는, t 개 이하의 결함 노드에 대해서는 위치가 정확히 파악될 수 있으며, t 개를 초과하고 u 개 이하의 결함 노드에 대해서는 결함 노드의 존재 여부만을 진단할 수 있음을 의미한다. 따라서 링에서 양방향으로 테스트가 이루어질 경우에는, 양쪽 이웃 노드에 의한 테스트 결과가 모두 1인 노드는 결함노드로 진단이 이루어지며, 2개 이상이면서 최대 $|R|-2$ 개의 결함 노드가 존재할 경우에는 결함 노드의 존재 여부를 파악할 수 있다.

Feng은 시스템-레벨의 하이퍼큐브를 진단하는 적응적 알고리즘으로서 HADA(Hypercube Adaptive Diagnosis Algorithm)와 IHADA(Improved HADA)를 제안하였다[1]. HADA, IHADA는 하이퍼큐브 시스템을 기본 구조로 설계된 알고리즘이지만, 링이 임베딩될 수 있는 모든 네트워크에 적용 가능한 알고리즘이다. Feng이 제안한 방법에 의하면 HADA, IHADA를 사용하여 진단을 수행할 경우, Armstrong과 Gray가 제안한 비적용적 진단 알고리즘[3]과 Vaidya와 Pradhan이 제안한 ASDA에 비하여 적은 진단비용이 필요하다. 그리고, 많은 수의 결함노드를 극복할 수 있는 확률적 진단 알고리즘[5]과 비교적 적은 진단 비용을 필요로 하는 Pessimistic 알고리즘[6]보다도 적은 진단 비용으로 시스템을 진단할 수 있다. Feng은 n -큐브 시스템 내에 n 개를 초과하는 결함노드가 존재할 경우, 진단의 가능 여부에 대한 문제를 $over-d$ 문제라고 정의하였다. 이전의 연구에서는 $over-d$ 문제를 해결하기 위하여 확률적인 방법을 사용한 반면 Feng의 방법은 결정적인 방법을 사용함으로써 보다 정확한 진단이 수행되었다[7].

HADA와 IHADA는 Vaidya와 Pradhan에 의해 증명된 링 구조의 진단 특성[5]과 Divide-and-Conquer방법을 이용하여 하이퍼큐브를 진단하는 알고리즘이다. 이 알고리즘들은 RGC(Reflected Gray Code)를 이용하여

링을 하이퍼큐브에 임베딩한 후에 하이퍼큐브 구조의 계층적인 특성과 RGC의 특성을 이용하여 최대 1개의 결합노드를 포함하는 서브링이 발견될 때까지 초기의 링을 분할해 나간다. 링의 $1-FL-(|R|-2)-FD$ 성질에 의하여, 분할하여 얻어진 서브링에 최대 1개의 결합노드가 존재한다면 서브링을 구성하는 모든 노드들에 대한 정확한 진단이 수행될 수 있다. 또한 진단알고리즘의 수행 중에는 새로운 결합노드가 발생하지 않는다는 가정 하에, 전 단계에서 이미 이루어진 테스트 결과들을 이용하여 시스템 전체에 대한 진단을 수행한다.

HADA, IHADA는 기존의 연구결과에 비하여 적은 비용을 사용하여 시스템에 대한 진단을 수행할 수 있는 반면, 링을 분할할 때 하이퍼큐브의 구조적 특성을 충분히 활용하지 못하였다. 즉, 각각의 노드의 번호를 $d_n d_{n-1} \dots d_2 d_1$ 으로 표현할 때, $d_i (1 \leq i \leq n)$ 의 값이 0인 노드들로 구성된 서브링과 d_i 의 값이 1인 노드들로 구성된 서브링으로 i 번째의 분할이 이루어진다. 그러나 하이퍼큐브의 구조적 특성에 의하여 n -큐브를 분할하여 2개의 $(n-1)$ -큐브로 만들 수 있는 방법은 n 가지가 있고, 이러한 n 가지의 분할방법 중 최대 1개의 결합노드를 포함하는 서브링을 보다 적은 단계를 거쳐서 구성할 수 있는 분할이 이루어진다면 전체적인 진단 비용을 줄일 수 있게 된다.

본 논문에서는 링의 분할 단계를 줄이기 위한 방법으로서 적응적 큐브 분할 방법을 제안한다. 적응적 큐브 분할 방법은 신드롬 분석을 통하여 비결함 노드들을 예측한 후, 예측결과를 이용하여 현재 단계에서 가능한 분할방법 중에서 최소의 결합노드를 포함한 서브링을 구성할 수 있는 분할을 수행한다. 따라서 최대 1개의 결합노드를 포함하는 서브링을 보다 적은 분할 단계를 통해서 구성할 수 있게되며, 이로 인하여 전체적인 진단 비용을 줄이게 된다. 시뮬레이션 결과는 본 논문에서 제안한 방법을 적용할 경우, 평균적으로 Feng이 제안한 방법에 비해 보다 적은 테스트 라운드와 테스트 링크를 통하여 진단이 이루어짐을 보여준다.

3. 적응적 큐브 분할을 이용한 하이퍼큐브 진단 알고리즘

이 논문에서 제안하는 알고리즘의 설명의 편의를 위해서 사용하는 기호들의 정의는 다음과 같다.

FF : n -큐브($n \geq 3$)를 구성하는 노드들 중, 비결함인 노드들로 구성된 집합.

FF_{guess} : n -큐브($n \geq 3$)를 구성하는 노드들 중, 비결함으로 예측된 노드들로 구성된 집합.

$SC0_i$: n -큐브($n \geq 3$)를 구성하는 노드의 노드번호를 $d_n d_{n-1} \dots d_2 d_1$ 와 같이 표현할 경우, d_i 의 값이 0인 노드들로 구성된 집합(단, $1 \leq i \leq n$).

$SC1_i$: n -큐브($n \geq 3$)를 구성하는 노드의 노드번호를 $d_n d_{n-1} \dots d_2 d_1$ 와 같이 표현할 경우, d_i 의 값이 1인 노드들로 구성된 집합(단, $1 \leq i \leq n$).

$FF0_i$: $FF0_i = FF_{guess} \cap SC0_i$ (단, $1 \leq i \leq n$).

$FF1_i$: $FF1_i = FF_{guess} \cap SC1_i$ (단, $1 \leq i \leq n$).

f_{cont} : 링 상에서 연속해서 인접한 결합노드의 수.

$MAX(f_{cont})$: 링 상에서 연속해서 인접한 결합노드의 최대 수.

본 논문에서 제안하는 진단 알고리즘의 대부분은 Feng이 제안한 방법과 유사하게 수행된다. 즉, RGC를 이용하여 하이퍼큐브에 링을 임베딩한 후 최대 1개 이하의 결합노드를 포함한 서브링이 발견될 때까지 divide-and-conquer방법에 의해 n -큐브를 $2 \times (n-1)$ -큐브로 분할해 나간다. 만약 최대 1개 이하의 결합노드를 포함한 서브큐브(서브링)가 발견되면, $1-FL-(|R|-2)-FD$ 와 같은 링구조의 진단 특성에 의해 서브링에 대한 정확한 진단이 이루어지며 이전 단계에서의 테스트 결과(신드롬)들을 이용하여 나머지 노드들에 대한 진단을 수행하게 된다.

그러나, HADA, IHADA에서 i 번째 분할이 d_i (노드번호: $d_n d_{n-1} \dots d_2 d_1$)에 의하여 정적인 방법으로 이루어지는 것과는 다르게, 본 논문에서 제안하는 알고리즘에서는 함수 $Find_Dimension_of_Partition()$ 의 결과(분할 차원)를 이용하여 n -큐브로부터 $2 \times (n-1)$ -큐브를 만들게 된다. $Find_Dimension_of_Partition()$ 는 n -큐브에서 가능한 n 가지의 분할 차원들 중에서 다음과 같이 두 과정을 거쳐서 유리한 분할 차원을 선택하게 된다.

1. 신드롬 분석: 하이퍼큐브에 매핑된 링에 대하여 시계방향의 테스트와 반시계방향의 테스트를 수행하고 난 후 얻어지는 신드롬에서 두 방향의 결과가 모두 비결함으로 나타난 노드들로 구성된 집합 FF_{guess} 를 만든다. 그런 후, FF_{guess} 의 원소들 중, 신드롬에서 시계방향 또는 반시계방향의 테스트에서 결함으로 나타난 노드와 인접한 노드를 FF_{guess} 로부터 제외시킨다.
2. 분할 차원 선택: n 개의 분할 방법 중에서, FF_{guess}

의 원소를 가장 많이 포함하는 서브링을 생성할 수 있는 분할 차원을 선택한다. 물론, 신드롬 분석을 통하여 비결합이 아닌 노드로 예측된 노드들의 집합 FF_{guess} 에 결합이 있는 노드가 포함될 가능성이 있다. 그러나, $MAX(f_{com}) \leq 4, n \geq 3$ 을 만족한다면 FF_{guess} 를 구성하는 노드는 모두 결합이 아니다(4장 보조정리1). 따라서, FF_{guess} 를 최대한 많이 포함하는 $(n-1)$ -큐브를 생성할 수 있는 분할방법을 선택함으로써 평균적으로 Feng의 방법보다 적은 단계를 거쳐서 최대 1개 이하의 결합노드를 포함하는 서브링을 찾게 된다.

본 논문에서 제안하는 하이퍼큐브 진단 알고리즘은 다음과 같다.

Diagnosis(n -큐브)

입력 : n -큐브

단계1 : RGC를 이용하여 n -큐브에 2^n 개의 노드로 구성된 링을 매핑한다.

단계2 : Ring_Diagnosis(n -큐브, 0)를 수행한 결과, 만약 결합노드의 수가 1개 이하이면 링 구조의 진단 특성상 진단을 정확히 수행할 수 있으므로 진단을 종료하고, 결합노드의 수가 1개 이하가 아니면 정확한 진단이 불가능하므로 단계3을 수행한다.

단계3 : 분할 차원을 선택하기 위하여 Find_Dimension_of_Partition(n -큐브)을 수행한다. Find_Dimension_of_Partition(n -큐브)의 실행결과 얻어지는 분할 차원을 d 라 할 경우, d 에 의한 분할과, 분할된 각 서브큐브에 대한 진단을 수행하기 위하여 Subcube_Diagnosis(n -큐브, d)를 수행한다.

Function Ring_Diagnosis(n -큐브, d)

입력 : n -큐브, 분할차원 d

단계1 : d 의 값이 0일 경우, 각 노드는 n -큐브에 매핑된 링 상에서 인접한 노드에 대하여 시계방향의 테스트를 수행한다. 만약 d 의 값이 0이 아닐 경우는 시계방향의 테스트를 수행하되, 인접한 노드에 대한 시계방향의 테스트를 이전 단계에서 이미 수행한 노드는 분할차원 d 에 의해 대응되는 노드(노드번호에서 분할차원의 비트값만이 서로 다른 노드)에 대한 테스트를 수행한다. 또한 각 노드의 테스트 결과(신드롬)는 저장한다. 시계방향의 테스트 결과가 모두 0으로 나타날 경우는 결합노드의 수 0을 반환하고, 그렇지 않을 경우는 단계2를 수행한다.

단계2 : 단계1과 동일한 방법으로 반시계방향의 테스트를 수행한 후, 각 노드의 테스트 결과(신드롬)는 저장한다.

단계3 : 단계1과 단계2를 수행한 결과 시계방향과 반시계방향의 테스트 결과가 1로 나타난 노드의 수가 1개 일 경우는 해당 노드를 결합노드로 표시하고 결합노드의 수 1을 반환한다. 만약 결합노드를 결정할 수 없으면 0과 1이 아닌 값을 반환한다.

Function Find_Dimension_of_Partition(n -큐브)

입력 : n -큐브

단계1 : n -큐브의 신드롬에서 시계방향과 반시계방향의 테스트 결과값이 모두 0으로 나타난 노드들로 집합 FF_{guess} 를 구성한 후, FF_{guess} 의 원소들 중 시계방향 또는 반시계방향의 테스트 결과가 1인 노드와 인접한 노드는 FF_{guess} 에서 제외시킨다.

단계2 : m_i, m_j 를 각각 $m_i = MAX(|FF0_i|, |FF1_i|), m_j = MAX_{1 \leq i \leq n}(m_i)$ 와 같이 정의할 경우(단, $1 \leq i \leq n, 1 \leq j \leq n$), m_j 의 인덱스값 j 를 분할차원으로 반환한다.

Function Subcube_Diagnosis(n -큐브, d)

입력 : n -큐브, 분할차원 d

단계1 : 분할차원 d 를 이용하여 n -큐브를 두 개의 $(n-1)$ -큐브로 분할한다.

단계2 : 각 $(n-1)$ -큐브에 대하여 Ring_Diagnosis($n-1, d$)를 수행한 결과, 만약 결합노드의 수가 1개 이하이면 진단을 정확히 수행할 수 있으므로 진단을 종료하고, 결합노드의 수가 1개 이하가 아니면 정확한 진단이 불가능하므로 단계3을 수행한다.

단계3 : 각 $(n-1)$ -큐브에 대하여 Find_Dimension_of_Partition()를 수행한다.

단계4 : 단계3의 결과를 이용하여 각 $(n-1)$ -큐브에 대하여 Subcube_Diagnosis()를 순환적으로 수행한다.

Ring_Diagnosis()에서는, 결합노드의 수가 $|R|-2$ 보다 작다는 가정에 의해, 시계방향의 테스트 결과가 모두 비결합인 것으로 나타날 경우 모든 노드를 비결합으로 정확히 진단할 수 있으므로 반시계방향의 테스트를 수행하지 않고 진단과정을 종료하게 된다. 또한, Subcube_Diagnosis()에서는 Divide-and-Conquer 방법에 의하여, n -큐브로부터 생성된 2개의 $(n-1)$ -큐브에 다시 서브링을 임베딩한 후 시계방향과 반시계방향의 테스트를 수행하게 된다. 이 경우 각 $(n-1)$ -큐브 내에서 시계방향과 반시계방향의 테스트가 수행될 때,

수행되어야 할 테스트가 이전 단계에서 이미 이루어진 노드들이 존재할 경우가 있는데, 병렬 처리의 극대화를 위하여 이와 같은 노드들은 분할 차원에 의해 대응되는 노드들에 대한 테스트를 수행한다. 이러한 테스트는 함수 $Ring_Dinagnosis()$ 의 단계1과 단계2에서 수행된다.

4. 알고리즘 분석

HADA와 비교하여 설명하기 위하여 HADA에서 사용한 진단 비용 모델을 이용한다. Feng은 진단비용의 측정을 위해 테스트 라운드와 테스트 링크를 사용하고 있으며, 하나의 노드가 테스트를 수행하는 중에 테스트를 받을 수 없다는 가정을 하고 있다. 따라서 하나의 링을 구성하는 각각의 노드들이 모두 테스트를 수행하기 위해서는 2번의 테스트 라운드가 필요하다. 이와 같은 가정을 기초로 하여 HADA에서 수행되는 테스트 라운드와 테스트 링크는 다음과 같다. 단, $TL_n(f_1, f_2)$ 와 $TR_n(f_1, f_2)$ 는 각각 HADA를 사용했을 경우 진단을 위해 사용되는 테스트 링크의 수와 테스트 라운드의 수를 나타내며, f_1, f_2 는 각각 n-큐브가 $2 \times (n-1)$ -큐브로 분할 되었을 경우 분할된 각 서브큐브에 포함된 결합노드를 나타낸다[1].

정리1. HADA에서 필요한 테스트 링크의 개수는 다음과 같다[1].:

$$TL_n(f_1, f_2) = \begin{cases} 2^n & f_1 = 0 \text{ and } f_2 = 0, 1 \\ 3 \times 2^{n-1} & f_1 = 0 \text{ and } 1 < f_2 \leq n, \text{ or} \\ & f_1 \text{ and } 1 \leq f_2 \leq n-1 \\ 3 \times 2^{n-1} + 1 & f_1 \text{ and } 1 < f_2 = n-1 \\ 2^{n-1} + TL_{n-1}(f_1, f_2) + TL_{n-1}(f'_1, f'_2) & f_1, f_2 > 1, 1 < f_1 + f_2 \leq n, \\ & f_1 + f_2 = f_1, \text{ and } f'_1 + f'_2 = n \end{cases}$$

이러한 진단에 필요한 시간은 다음과 같다[1].:

$$TR_n(f_1, f_2) = \begin{cases} 2 & f_1, f_2 = 0 \\ 4 & f_1 = 0 \text{ and } f_2 = 1 \\ 5 & f_1 = 0 \text{ and } 1 < f_2 \leq n \\ 6 & f_1 = 1 \text{ and } f_2 = 1 \\ 7 & f_1 = 1 \text{ and } 1 < f_2 \leq n-1 \\ 2 + \max(TR_{n-1}(f_1, f_2), TR_{n-1}(f'_1, f'_2)) & f_1, f_2 > 1, 1 < f_1 + f_2 \leq n, \\ & f_1 + f_2 = f_1 \text{ and } f'_1 + f'_2 = n \end{cases}$$

본 논문에서 제안한 방법에 의하면, 우선 분할된 서브 큐브에서의 결합노드의 분포가 다음과 같은 경우에 대해서는 HADA에서와 같은 진단 비용과 테스트 링크를 필요로 하게 된다.

$$\begin{aligned} f_1, f_2 &= 0, \\ f_1 &= 0 \text{ and } f_2 = 1, \\ f_1 &= 0 \text{ and } 1 < f_2 \leq n \end{aligned}$$

반면 다음의 경우에 있어서는 결합노드의 분포에 따라서 HADA 보다 작거나 같은 진단 비용과 테스트 링

크를 사용하여 진단할 수 있다.

$$\begin{aligned} f_1 &= 1 \text{ and } f_2 = 1, \\ f_1 &= 1 \text{ and } 1 < f_2 \leq n-1, \\ f_1, f_2 &> 1, 1 < f_1 + f_2 \leq n, f_1 + f_2 = f_1 \text{ and } f'_1 + f'_2 = n \end{aligned}$$

보조정리1. n-큐브가 $MAX(f_{cont}) \leq 4, n \geq 3$ 를 만족하면, $Find_Dimension_of_Partition()$ 에서 비결함노드로 예측되는 노드들의 집합 FF_{guess} 는 다음을 만족한다.

$FF_{guess} \subset FF$, 단 FF 는 하이퍼큐브 내의 비결함노드의 집합을 나타낸다.

증명. 다음과 같이 $MAX(f_{cont})$ 가 1, 2, 3, 4인 경우로 나누어 생각해 보자.

Case1. $MAX(f_{cont}) = 1, n \geq 2$.

결함이 있는 장치의 시계방향으로 인접된 비결함노드과 반시계방향으로 인접된 비결함노드에 의하여 양방향의 테스트 결과가 모두 1이 되므로 $Find_Dimension_of_Partition()$ 의 단계1에 의하여 결합노드가 FF_{guess} 로부터 제외된다. 따라서, $FF_{guess} \subset FF$ 를 만족한다.

Case2. $MAX(f_{cont}) = 2, n \geq 3$.

연속된 두개의 결합노드는 각각 1개 씩의 인접한 비결함노드를 가지게 된다. 따라서 결합노드들은 이러한 인접된 비결함노드에 의하여 적어도 한 방향의 테스트 결과가 1이 되므로 단계1에 의하여 FF_{guess} 로부터 제외된다. 따라서 두개의 모든 결합노드들이 FF_{guess} 로부터 제외되므로 $FF_{guess} \subset FF$ 를 만족한다.

Case3. $MAX(f_{cont}) = 3, n \geq 3$.

연속된 세개의 결합노드들 중 두 개의 결합노드들은 각각 1개 씩의 비결함노드들과 인접하게 된다. 이렇게 비결함노드들과 인접된 두 개의 결합노드들은 적어도 한 방향의 테스트 결과가 1이 되므로 단계1에 의하여 FF_{guess} 로부터 제외된다. 또한 결합노드들로 둘러싸인 하나의 결합노드는 인접된 노드들이 모두 단계1에 의하여 결합노드들로 예측되었으므로, FF_{guess} 로부터 제외된다. 따라서 세 개의 모든 결합노드들이 FF_{guess} 로부터 제외되므로 $FF_{guess} \subset FF$ 를 만족하게 된다.

Case4. $MAX(f_{cont}) = 4, n \geq 3$.

연속된 네개의 결합노드들 중 두개의 결합노드들은 각각 1개 씩의 비결함노드와 인접하게 된다. 이렇게 비결함노드와 인접된 두개의 노드는 적어도 한 방향의 테스트 결과가 1이 되므로 단계1에 의하여 FF_{guess} 로부터 제외된다. 또한 결합노드들로 둘러싸인 두 개의 결합노드들은 단계1에 의하여 결합노드므로 예측된 두개의 노드

들과 각각 인접해 있으므로, FF_{guess} 로부터 제외된다. 따라서 네 개의 모든 결합노드들이 FF_{guess} 로부터 제외되므로 $FF_{guess} \subset FF$ 를 만족하게 된다. □

HADA방법에 의하면, 최대 1개의 결합노드를 포함하는 서브큐브를 찾을 때까지 분할이 이루어지며 i 번째 단계의 분할이 d_i (노드번호: $d_n, d_{n-1}, \dots, d_2, d_1$)에 의해 이루어진다. 그러나, 본 논문에서 제안한 방법은, 위에 제시한 보조정리1을 이용한 $Find_Dimension_of_Partition()$ 에 의해서, n -큐브에서 가능한 n 가지의 분할 방법 중 비결함인 것으로 예측된 노드들의 집합인 FF_{guess} 의 원소를 가장 많이 포함하도록 하는 분할 방법(분할 비트)을 선택하게 된다. 즉, 분할된 두 개의 서브큐브에서 결합노드가 가능한 한 한쪽 서브큐브에 모이도록 하는 분할을 선택하게 된다. 또한, 본 논문에서 제안하는 알고리즘은 정리2에 제시된 조건이 만족되는 상황에서는 단 한번의 분할을 통하여 시스템 전체에 대한 진단을 정확히 수행할 수 있다.

정리2. $MAX(f_{con}) \leq 4, n \geq 3$ 를 만족하는 n -큐브의 분할에 있어서, $|FF_{guess}| \geq 2^{n-1}$ 이고 FF_{guess} 의 부분집합으로 $(n-1)$ -큐브를 구성할 수 있는 분할 비트가 존재한다면, 1번의 분할에 의해 비결함노드로만 구성된 $(n-1)$ -큐브를 구성할 수 있다.

증명. n -큐브에서의 노드번호를 $d_n, d_{n-1}, \dots, d_2, d_1$ 와 같이 표현하면, 임의의 분할 비트 d_i 를 선택하여 d_i 가 0인 $(n-1)$ -큐브와 1인 $(n-1)$ -큐브를 구성할 수 있다. 이 경우 임의의 분할 비트 d_i 에 의해 생성되는 두개의 $(n-1)$ -큐브를 각각 SC_0, SC_1 와 같이 표현하자. 이 때 각각의 $(n-1)$ -큐브를 구성하는 노드의 수는 2^{n-1} 가 되며, 이때 생성될 수 있는 $(n-1)$ -큐브의 총수는 $1 \leq i \leq n$ 에 의하여 $2 \times n$ 이 된다. 만약 $|FF_{guess}| \geq 2^{n-1}$ 이고, $2 \times n$ 개의 $(n-1)$ -큐브 중 FF_{guess} 의 부분집합으로 구성할 수 있는 $(n-1)$ -큐브, SC_* (단, $*$ 는 0 또는 1)가 존재한다면, 보조정리1에 의하여 SC_* 를 구성하는 노드는 모두 결함이 아닌 노드이다. 즉, SC_* 의 구성 노드는 모두 결함이 아니므로 정확한 진단을 수행할 수 있다[5]. 또한 분할 비트 d_i 에 의해 생성되는 두개의 $(n-1)$ -큐브, SC_0, SC_1 간에는 d_i 에 의해 대응되는 노드사이에 2^{n-1} 개의 간선들이 존재하게 된다. 따라서, 정확한 진단이 수행된 SC_* 는 이와 같은 간선을 통하여 대응 서브큐브

에 대한 진단을 정확히 수행할 수 있다. □

정리3. n -큐브를 구성하는 모든 노드의 결합 발생 확률이 동일하며 총 f 개의 결합노드가 존재한다고 가정할 경우, 1번의 분할에 의해 비결함노드로만 구성된 $(n-1)$ -큐브를 구성할 수 있는 분할 비트가 존재할 확률은 $2^n \prod_{i=0}^{f-1} \frac{(2^{n-1}-i)}{(2^n-i)}$ 이다.

증명. n -큐브로부터 1번의 분할에 의해 비결함노드로만 구성된 $(n-1)$ -큐브를 구성할 수 있는 분할 비트가 존재할 확률은 f 개의 결합노드가 n -큐브로부터 생성 가능한 $(n-1)$ -큐브들 중 하나의 큐브에 모두 포함될 확률과 동일하며 다음과 같은 식에 의해 구할 수 있다.

$$\frac{(n\text{-큐브로부터 생성될 수 있는 } (n-1)\text{-큐브의 총 수}) \times ((n-1)\text{-큐브에 } f\text{개의 결합노드가 모두 존재하는 경우의 수})}{(n\text{-큐브에 } f\text{개의 결합노드가 존재하는 경우의 수})} \quad (1)$$

우선, n -큐브에 f 개의 결합노드가 존재하는 경우의 수는 2^n 개의 노드 중 f 개의 결합노드를 선택하는 조합으로 볼 수 있으므로 다음과 같이 나타낼 수 있다.

$$(n\text{-큐브에 } f\text{개의 결합노드가 존재하는 경우의 수}) = \prod_{i=0}^{f-1} \frac{(2^n-i)}{(f-i)}$$

또한, $(n-1)$ -큐브에 f 개의 결합노드가 존재하는 경우의 수는 2^{n-1} 개의 노드 중 f 개의 결합노드를 선택하는 조합으로 볼 수 있으므로 다음과 같이 나타낼 수 있다.

$$((n-1)\text{-큐브에 } f\text{개의 결합노드가 존재하는 경우의 수}) = \prod_{i=0}^{f-1} \frac{(2^{n-1}-i)}{(f-i)}$$

따라서, n -큐브로부터 생성될 수 있는 $(n-1)$ -큐브의 총 수는 $2 \times n$ 임으로 구하고자 하는 확률은 다음과 같다.

$$2 \times n \times \prod_{i=0}^{f-1} \frac{2^{n-1}-i}{2^n-i} \quad \square$$

정리4. n -큐브를 구성하는 모든 노드의 결합 발생 확률이 동일하며 총 f 개의 결합노드가 존재한다고 가정할 경우, HADA에 의해 1번의 분할에 의해 비결함노드로만 구성된 $(n-1)$ -큐브를 구성할 수 있는 분할 비트가 존재할 확률은 $2 \prod_{i=0}^{f-1} \frac{(2^{n-1}-i)}{(2^n-i)}$ 이다.

증명. HADA는 각 분할단계의 분할 비트가 정해져 있으므로 n -큐브로부터 생성될 수 있는 $(n-1)$ -큐브의 총 수는 2가 된다. 나머지 증명 과정은 정리3의 증명과 동일하다. □

정리2,3,4에 의해서, 모든 노드의 결합 발생 확률이 동일하다고 가정할 경우, 제안된 알고리즘은 한번의 분할을 수행했을 경우 분할된 2개의 서브큐브에 포함된

결합노드의 수가 각각 0과 f (단, f 는 전체 결합노드의 수)가 될 확률이 HADA에 비하여 n 배 큰 것을 알 수 있다.

5. 시뮬레이션 결과

이 논문에서 제안하는 알고리즘의 진단 비용을 살펴 보기 위하여 다음과 같이 시뮬레이션을 수행하였다. 시뮬레이션에 대한 가정은 다음과 같다.

- (1) 하이퍼큐브를 구성하는 모든 노드들의 결합 발생 확률은 동일하다.
- (2) 결합노드는 임의의 테스트 결과를 나타낼 수 있으며 테스트 결과가 옳은 경우와 그른 경우의 확률은 동일하다.
- (3) 적응적 분할과정을 포함한 진단과정은 신드롬 분석을 담당하는 별도의 프로세서에 의해 진행된다.

시뮬레이션은 다음과 같은 방법으로 수행하였다.

- (1) 진단 알고리즘은 C언어를 이용하여 PC상에서 구현하였으며, 각각의 노드는 인접 노드에 대한 테스트 링크 역할을 하는 변수를 가진 구조체 형태의 데이터 구조로 표현하였다.
- (2) 10000가지의 결합노드 패턴을 무작위로 생성한 후, 동일한 결합노드 패턴에 대하여 Feng의 알고리즘과 이 논문에서 제안한 방법을 각각 적용하였다.
- (3) Feng의 알고리즘과 이 논문에서 제안한 알고리즘은 모두 순환적인 특성을 가지고 있으므로 1단계의 분할만 수행한 후 결합노드의 분포 변화에 대한 경우의 수를 측정하였다.

표 1 $f=n$ 인 n -큐브에서 1단계 이하의 분할을 통한 시스템 진단이 가능한 경우의 수

차원의 수	HADA			Proposed Algorithm		
	0 : f	1 : $f-1$	1단계 분할로 진단 가능한 경우	0 : f	1 : $f-1$	1단계 분할로 진단 가능한 경우
4	793	4848	5641	2197	5641	7838
5	429	2883	3312	1542	5520	7062
6	229	1761	1990	1005	4932	5937
7	145	1027	1172	719	3995	4714
8	52	589	641	408	2930	3338
9	21	272	293	253	2164	2417
10	4	145	149	129	1464	1593

- (3) 하이퍼큐브의 차원의 수(n)과 결합노드의 수(f)가 같다는 가정 하에 하이퍼큐브의 차원 수를 4에서 10까지 변화시키며 측정하였다[표1, 그림1,2].
- (4) 6-큐브에서 결합노드의 수(f)를 2에서 9까지 변화시키며 측정하였다[표2, 그림3,4].

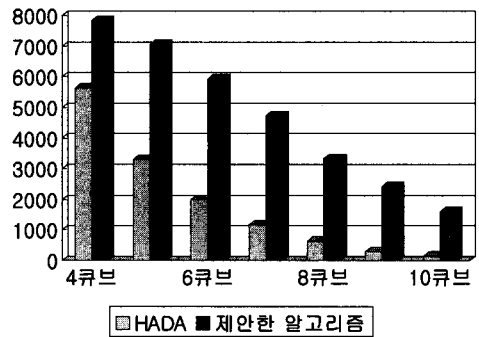


그림 1 표1에 대한 그래프

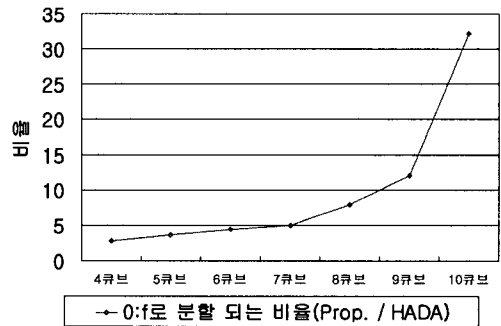


그림 2 표1에서 1번 분할 후 0:f로 분할되는 비율

다음의 표1은 결합노드의 수를 f 라 할 경우, $f=n$ 을 만족하는 결합 노드가 존재하는 n -큐브에 대하여 Feng 이 제안한 방법을 적용했을 경우와 본 논문에서 제안한 방법을 적용했을 경우, 각각에 대하여 1번의 분할 후 두 개의 $(n-1)$ -큐브에 대한 결합노드의 분포가 각각 0 : f 와 1 : $f-1$ 이 되는 경우의 수를 나타낸다. 즉 1번 이하의 분할을 통하여 시스템 전체에 대한 진단이 가능한 경우의 수를 나타낸다.

위의 표1에 나타난 결과와 그림1,2로부터, 큐브의 차원에 관계없이, 제안한 분할 방법을 사용하면 1번 이하의 분할을 통하여 시스템을 진단할 수 있는 경우가 Feng의 방법에 의한 것보다 많다는 것을 확인할 수 있

다. 또한, 그림2에 나타난 것과 같이 1단계 분할 후 결합노드의 분포가 0 : f와 같이 나타나는 경우만을 고려할 경우 그 비율은 큐브의 차원이 수가 증가함에 따라 큰 차이를 보였다. 따라서, $f = n$ 인 경우, 제안한 방법을 적용하면 본 논문에서 제안한 방법이 Feng의 방법보다 많은 경우의 결합노드 분포 상태에 대하여 적은 테스트 단계를 통하여 시스템을 진단할 수 있음을 확인할 수 있다.

표 2 6-큐브에서 1단계 이하의 분할을 통한 시스템 진단이 가능한 경우의 수

결합 노드의 수 f	HADA			Proposed Algorithm		
	0 : f	1 : f-1	1단계 분할로 진단 가능한 경우	0 : f	1 : f-1	1단계 분할로 진단 가능한 경우
2	4924	5076	10000	9676	324	10000
3	2362	7638	10000	7254	2746	10000
4	1122	4986	6108	4222	5463	9685
5	530	3043	3573	2172	6070	8242
6	239	1782	2021	1041	4946	5987
7	89	920	1009	487	3248	3735
8	40	491	531	197	1888	2085
9	10	203	213	89	1054	1143

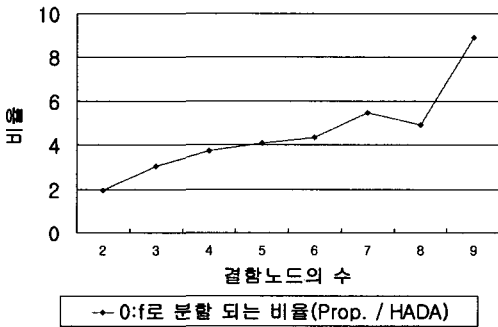


그림 4 표2에서 1번 분할 후 0:f로 분할되는 비율

다음의 표2와 그림3,4는 6-큐브에서 포함된 결합노드의 수의 변화에 따른 시뮬레이션의 결과이다. 위의 시뮬레이션에서와 마찬가지로 1단계 분할을 통한 결합 노드의 분포 변화를 관찰하였다. 결과에 의하면 다양한 결합 노드의 수에 대해서도 제안한 방법이 Feng의 방법보다

많은 경우의 결합노드 분포 상태에 대하여 적은 테스트 단계를 통하여 시스템을 진단할 수 있음을 확인할 수 있다.

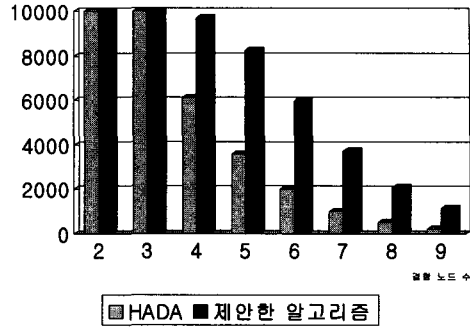


그림 3 표2에 대한 그래프

6. 결론

시스템-레벨 진단 알고리즘은 병렬시스템에서 중요한 연구분야이다. 또한 하이퍼큐브 구조는 계층적이며 정규적인 특성을 가지고 있으며 최근들어 병렬처리 시스템에 많이 이용되고 있다. Feng [1]은 하이퍼큐브구조의 병렬시스템의 진단알고리즘으로써 HADA와 IHADA를 제안하였으며 이 방법은 기존의 연구결과보다 [2, 3] 적은 진단 비용으로 시스템을 진단할 수 있었다. HADA와 IHADA에서는 정적인 패턴의 큐브 분할 단계를 거쳐서 시스템 진단이 수행되었다. 본 논문에서 제안한 적응적 큐브 분할 방법은 신드롬을 분석을 통하여 다음 단계에 유리한 분할 방법을 적응적으로 선택하여 큐브를 분할하였다. 즉, 결합노드 분포의 경우의 수에 대하여, Feng의 알고리즘에 비하여 보다 많은 경우에 대하여 한번의 분할을 통해 비결합노드로 구성된 서브큐브를 구성함으로써 전체적인 진단비용을 줄일 수 있다. 또한 시뮬레이션을 통하여 다양한 결합 노드들의 분포에 대하여 본 논문에서 제안한 방법이 Feng에 의해 제안된 방법에 비하여 많은 경우에 적은 진단 비용과 테스트를 통하여 시스템을 진단할 수 있음을 확인할 수 있었다. 앞으로의 연구방향은 본 논문에서 제안한 방법보다 많은 수의 비결합노드를 정확하게 예측하는 방법을 연구함으로써 분할 차원의 선택의 신뢰성을 높이는 것이다. 또한 시스템-레벨의 가정은 많은 제약점을 가지고 있으므로 분산적으로 수행되는 진단알고리즘에 대한 연구도 요구된다.

참 고 문 헌

- [1] C. Feng, L. N. Bhuyan, and F. Lombardi, "Adaptive System-Level Diagnosis for Hypercube Multiprocessors," IEEE Trans. Computers, vol. 45, pp. 1137-1170, Oct. 1996.
- [2] F. P. Preparata, G. Metze, and R. T. Chien, "On the Connection Assignment Problem of Diagnosable Systems," IEEE Trans. Electronic Computers, no. 12, pp.848-854, Dec.1967.
- [3] J.Armstrong and F.Gray, "Fault Diagnosis in a Boolean n-cube Array of Microprocessors," IEEE Trans. Computers, vol.30, no.8, pp.587-590, Aug. 1981.
- [4] N. Nakajima, "A New Approach to System Diagnosis," Proc. 19th Allertion Conf. Comm., Control, and Computing, pp. 697-706, 1981.
- [5] Nitin H.Vaidya and Dhiraj K.Predhan, "Safe System Level Diagnosis," IEEE Trans. Computers, vol. 43, No. 3, Mar.1994.
- [6] A.Kavianpour and K.H.Kim, "A Comparative Evaluation of Four Basic System-Level Diagnosis Strategies for Hypercubes," IEEE Trans. Reliability, no.1, pp.26-37, Mar.1992.
- [7] F. J. Meyer and D. K. Pradhan, "Dynamic Testing Strategy for Distributed Systems," IEEE Trans. Computers, vol. 38, no. 3, pp 356-365, Mar.1989.
- [8] D. P. Bertsekas and J. N. Tsitsiklis, Parallel and Distributes Computation, Numerical Methods. Prentice-Hall Int'l, 1989.



최 문 옥

1998년 충북대학교 컴퓨터과학과 학사.
2000년 충북대학교 전자계산학과 석사.
관심분야는 결함허용, 병렬분산처리, 데이터마이닝 등



이 충 세

1989년 University of South Carolins, 전산학 박사. University of North Dakota, 전산학과 조교수. 동아대학교 경영정보학과 부교수. 현재 충북대학교 컴퓨터과학과 교수. 관심분야는 결함허용, 알고리즘 및 전문가 시스템