論文94-31B-2-1

분산환경에서의 데이타 가용성 향상을 위한 HQC 구조의 재구성 방법

(A Reconstruction Method of HQC structure for Improving Availability of Data in Distributed Environment)

劉憲昌*, 趙東榮**, 孫進坤***, 黃鍾善*

(Heon Chang Yu, Dong Young Cho, Jin Gon Shon and Chong Sun Hwang)

要約

분산 환경에서 데이타의 중복은 데이타의 가용성(availability)을 증가시키고 통신비용을 감소시키는 장점이 있다. 그러나 지역이상(site failure)이 발생할 경우 데이타의 일관성(consistency) 유지가 어려워지고 가용성이 떨어지는 문제가 생긴다. 특히 데이타의 일관성을 유지하기 위해 기존의 계층적 정족수 동의(Hierarchical Quorum Consensus: HQC)기법을 이용하는 경우 지역이상의 발생은 충분한 정족수를 얻지 못해 분산된 중복데이타에 대한 연산이 용이하지 않도록 만든다. 본 논문은 지역이상이 발생하였을 때, HQC 구조의 재구성을 통해 보우팅에서 동의를 얻을 수 있는 가능성을 높이는데 목적을 두고 있다. 그리고 기존의 정족수 동의기법과 HQC 기법에 대해 HQC 구조의 재구성에 따른 가용성 향상을 비교평가한다.

Abstract

In distributed environments, data replication increases availability and decreases communication cost. However, it is difficult to maintain consistency and availability of data if site failure occurs. When we use the conventional hierarchical quorum consensus (HQC) method in order to maintain the consistency of data, occurrence of site failures makes it harder to perform the operation on replicated data because of insufficient votes. The objective of this paper is to improve the possibility of retaining necessary votes by reconstructing the HQC structure, when the site failure occurs. Furthermore, we compare the modified HQC method with the conventional HQC and QC methods in terms of improvement of availability.

*正會員, 高麗大學校 電算科學科
(Dept. of Computer Science, Korea Univ.)
**正會員, 全州大學校 電子計算學科
(Dept. of Computer Science, Jeonjoo Univ.)
***正會員, 韓國放送通信大學校 電子計算學科
(Dept. of Computer Science, Korea Air and Correspondence Univ.)
接受日字 : 1993年 8月 19日

Ⅰ. 서론

최근들어 여러 지역에 분산되어 있는 개인용 워크스테이션의 사용자들이 전산망을 통해 지역적으로 떨어져 있는 정보들을 공유할 수 있도록 허용하는 분산시스템(distributed system)에 관심을 갖게 되었다. 이러한 분산시스템은 기존의 중앙집중식 시스템보다 두가지 측면에서 더 효율적이다. 첫째는 모든 기능이 여러 곳에 중복되어 있기 때문에 더 신뢰성이 있다.

둘째로, 분산시스템은 많은 연산이 병렬로 처리될 수 있기 때문에 같은 단위 시간당 더 많은 작업을 처리할 수 있다. 이와 같은 두가지 속성을 각각 결함허용(fault tolerance)과 병렬성(parallelism)이라 한다.[1,3,5,9]

이와 같은 분산시스템의 설계시 고려되어야 하는 요소들은 다음과 같다. 첫째는 여러 지역에 분산되어 있는 중복데이타에 대한 일관성(consistency)을 유지할 수 있어야 하고, 둘째는 이러한 중복데이타에 대한 높은 신뢰성(reliability) 및 가용성(availability)을 유지해야 하고, 세째는 공유하고 있는 중복데이타에 대한 동시접근을 조정해야 한다는 것이다.

여러 지역에 분산되어 중복된 데이타의 상호 접근을 할 때, 중복데이타를 가진 지역에 이상(failure)이 발생한다면 각 지역에 중복된 데이타들의 일관성 유지가 어렵고 가용성이 떨어지게된다. 이와같이 지역이상(site failure)이 발생하였을 경우의 문제에 대한 해결방법을 모색하고자 함이 본 연구의 동기가 되었다. 본 논문에서의 일관성이란 중복데이타를 가진 지역에 대한 중복데이타값의 동일성 유지 여부를 의미하고, 가용성이란 중복데이타에 대한 연산의 수행가능성을 의미한다.

이와같은 일관성 및 가용성을 유지하기 위하여 여러가지 기법들이 제안되어 왔다.[2] 본 연구는 여러가지 기법 중에서 정족수 동의기법과 관련하여 수행한다.

따라서 본 논문에서는 정족수 동의기법 중 계층적 정족수 동의기법의 문제점인 지역이상의 발생에 따라 가용성이 떨어지는 것을 막고, 가용성을 유지시켜 주기 위해 HQC 구조의 재구성 방법을 사용하는 개선된 계층적 정족수 동의기법을 제안한다.

Ⅱ. 정족수 동의 기법

데이타의 중복으로 인해 고려해야 할 문제는 중복데이타를 항상 똑같이 유지해야 하는 일관성(consistency) 문제인데, 이러한 일관성 문제를 해결하기 위해 여러 가지 기법들이 연구되어 왔으나, 이 장에서는 정족수 동의기법들에 대해 설명한다.

1. 정적/동적 정족수 동의기법

중복데이타의 동시성 제어를 위해 잘 알려진 알고리즘으로 Thomas와 Gifford에 의해 제안된 정적 정족수 동의(static quorum consensus)기법이 있다.[4,10] 이 기법은 중복데이타를 가진 지역에 보우팅 권한을 주어 할당된 정족수 만큼의 권한을 얻으면 읽기/쓰기 연산을 수행하도록 한다. 이때 중복데이타를 가진 지역의 전체수를 L, 읽기 연산을 위해 필요한 정족수를 r, 그리고 쓰기 연산을 위해 필요한 정족수를 w라 하면 각각에 대해 다음과 같은 관계를 유지한다

$$r + w > L$$
$$2w > L$$

정적 정족수 동의기법보다 좀더 높은 데이타의 가용성을 위해 Jajodia와 Mutchler가 제안한 동적 정족수 동의(dynamic quorum consensus) 기법은 데이타의 갱신횟수를 나타내는 버젼번호(version number)를 사용한다.[6] 즉 현재의 버젼번호는 중복데이타를 가지고 있는 모든 지역 중에서 가장 높은 버젼번호를 나타내는데 현재의 버젼번호를 가진 지역들에만 보우팅 권한을 주어 필요한 연산을 수행하는 기법이다.

2. 계층적 정족수 동의기법

정족수 동의기법의 문제점은 중복데이타를 가지고 있는 전체 지역의 수가 증가하면 할수록 주어진 연산을 수행하는데 필요한 정족수의 크기도 선형으로 증가한다는 것이다. 예를 들어 중복데이타를 가지고 있는 전체 지역의 수가 100개라 할 때 쓰기연산을 수행하기 위해서는 적어도 51개 지역으로 부터 동의를 얻어야 한다. 따라서 계층적 정족수 동의(hierarchical quorum consensus)기법은 동의에 필요한 정족수의 크기를 줄여보고자 하는 기법으로 A.Kumar에 의해 제안되었다.[7,8]

이 기법은 보우팅 권한을 갖는 지역들을 여러 소집단(subgroup)으로 나누어 각 지역의 동의를 거치고 그 다음으로는 소집단의 동의를 거치는 계층구조를 형성한다. 즉 9개의 지역이 보우팅 권한을 갖는다고 할 때, 3개 지역을 하나의 소집단으로 하여 3개의 소집단을 구성하고 데이타 갱신에 필요한 동의를 얻기

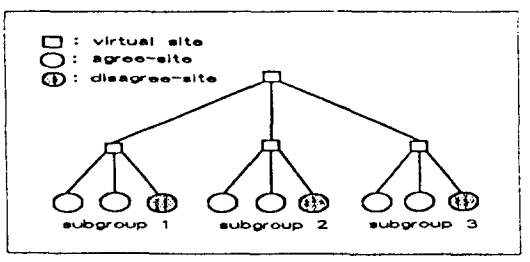


그림 1. 계층적 정족수 동의기법의 예
Fig. 1. Example of hierarchical quorum consensus method.

위해 결과적으로 2개의 소집단으로 부터 동의를 얻으면 되고 각 소집단으로부터는 2개의 지역으로 부터 동의를 얻으면 된다. 그러므로 전체적으로 동의를 얻는데 필요한 지역의 최소수는 4 개이다. 정족수 동의기법에서 5개 지역으로 부터 동의를 얻는 경우와 비교해 볼 때, 동의에 필요한 정족수의 크기가 줄어든다는 것을 알 수 있다.

그림 1에서 agree-site와 disagree-site는 각각 보우팅에 동의하는 지역과 동의하지 않는 지역을 나타낸다. 그리고 가상지역(virtual site)은 최하위레벨에서 소집단을 구성하는 지역들 중 선택된 임의의 한 지역으로 상위레벨의 소집단을 구성하는 한 원소이다.

Ⅲ. 계층적 정족수 동의기법의 문제점

Thomas와 Gifford에 의해 제안된 정적 정족수 동의기법에 비해 계층적 정족수 동의기법은 보우팅 과정에서 요구되는 정족수의 크기를 줄일 수 있다는 장점을 갖는다.[7,8] 따라서 지역이상이 발생하지 않는 정상적인 경우에는 상당한 성능향상이 있게 된다. 그러나 지역이상이 발생되는 경우, 어떠한 정족수 동의기법도 중복데이타에 대한 가용성의 저하를 막지 못한다. 계층적 정족수 동의기법에서 지역이상의 발생에 따른 가용성의 변화는 소집단들의 구성형태에 따라 달라진다. 본장에서는 계층적 정족수 동의기법의 이러한 가용성 특성이 가지는 문제점을 정족수 동의기법과 비교하여 설명한다

[정의 1]

S는 중복데이타를 포함하는 지역들의 전체집합이라고 하고, 계층적 정족수 동의기법에 의해, S가 동일한 갯수(m)의 지역들을 포함하는 n개의 소집단 $SG_1, \cdots, SG_n$ 으로 분할된다고 하자. 그러면 소집단들 사이의 지역이상의 발생분포를 나타내는 지역이상 분포벡터 SFj는 다음과 같이 정의된다

$$SFji = (f1, f2, \cdots, fn)$$

여기서, f_k : 소집단 SG_k에서 지역이상 빈도수

$(k=1, \cdots, n)$

$(0 \le f_1 \le f_2 \le \cdots \le f_n \le m)$

i : 전체 소집단에서 발생한 지역이상의 수.

즉, $i = f_1+f_2+\cdots+f_n$

j : i개의 지역이상이 발생한 경우 가능한 모든 경우들에 대한 인덱스

이제 9개의 중복데이타를 가진 지역들의 경우, 계층적 정족수 동의기법에서 발생되는 문제점을 CASE 1과 2에서 알아보도록 하자

CASE 1 4개의 지역이상이 발생한 경우

일반적인 정족수 동의기법에서 갱신연산을 수행하는데 필요한 정족수는 5 이므로 이 경우 보우팅과정에서 동의만 얻으면 아무 문제없이 갱신연산의 수행이 가능하다. 한편, 계층적 정족수 동의기법을 사용할 경우, 가능한 지역이상 분포벡터들은 다음과 같다

(1) $SF_1^4 = (0, 1, 3)$

(2) $SF_2^4 = (1, 1, 2)$

(3) $SF_3^4 = (0, 2, 2)$

(1)과 (2)의 경우는 보우팅과정을 수행할 때 소집단 1과 2가 가용하므로, 동의만 얻으면 갱신연산의 수행이 가능하다. 그러나, (3)의 경우는 소집단 2와 3의 지역이상 발생수가 각각 2이므로 소집단 2와 3은 가용하지 않다. 따라서 소집단 1로부터 동의를 얻더라도 소집단 2 와 3이 가용하지 않으므로 보우팅과정에서 동의를 얻을 수 없다. (3)의 경우, 전체적으로 5 개 지역이 가용함에도 불구하고 보우팅과정이 수행될 수 없게 된다

CASE 2 5개의 지역이상이 발생한 경우

지역이상의 발생 지역수가 5인 경우, 가능한 지역이상 분포벡터들은 다음과 같다.

(1) $SF_1^5 = (1, 1, 3)$

(2) $SF_2^5 = (1, 2, 2)$

(3) $SF_3^5 = (0, 2, 3)$

(1)의 경우, 소집단 1과 2가 가용하므로 보우팅과정에서 동의만 얻으면 갱신연산의 수행이 가능하다. 그러나, (2)와 (3)의 경우, 소집단 2와 3은 가용하지 않다. (2)와 (3)의 경우는 전체적으로 보우팅과정의 수행에 필요한 4개 지역이 가용함에도 불구하고, 두 소집단이 가용하지 않기 때문에 보우팅과정을 수행할 수 없다.

앞의 예에서 살펴본 바와 같이, 계층적 정족수 동의기법은 전체적으로 지역이상의 발생빈도가 같더라도, 전체 지역집합의 분할형태에 따라 다른 가용성의 결과를 초래한다는 문제점을 갖는다

Ⅳ. HQC 구조의 재구성 방법

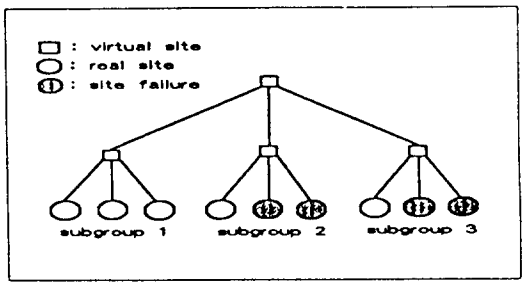
1. HQC 구조의 재구성

계층적 정족수 동의기법은 CASE 1과 CASE 2 에서와 같이 지역이상의 수가 동일하더라도 소집단들 사이의 지역이상의 발생분포에 따라 보우팅과정의 수행이 영향을 받는다는 문제점이 있었다. 이러한 문제점을 해결하기 위해서는 주어진 지역이상 분포벡터가 보우팅과정의 수행에 필요한 가용성을 갖지 못할 경

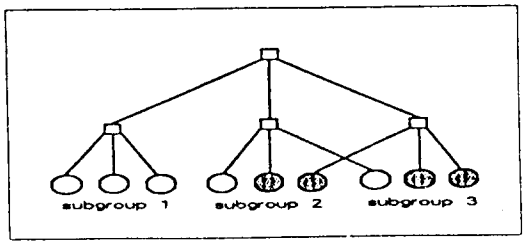
우, 소집단들 사이의 지역교환을 통해 소집단들을 논리적으로 재구성함으로써, 지역이상 분포벡터를 가용한 분포벡터를 갖도록 변경시키면 된다. 즉, 보우팅 과정의 수행에 필요한 가용성을 나타내는 지역이상 분포벡터를 얻도록 HQC 구조를 재구성하면 된다. CASE 1과 CASE 2의 경우, HQC 구조의 재구성과정은 다음과 같다

CASE 3 CASE 1의 (3) 경우

CASE 1의 (3) 경우는 소집단 2와 3이 가용하지 않으므로 보우팅과정이 수행되지 못했다 SF_3^1의 가용하지 않은 소집단 2와 3에서 비지역이상-노드의 전체 갯수는 2개로서 한 소집단 내에서 동의에 필요한 최소지역수 이상이다. 따라서, 소집단 2 혹은 3을 가용하도록 HQC 구조를 재구성한다면, 보우팅과정은 수행될 수 있게 된다. 그림 2에서와 같이 소집단 2의 지역이상-노드와 소집단 3의 비지역이상-노드를 교환하여 지역이상 분포벡터를 SF_1^4로 변경시키면 된다(그림 2). SF_3^1에서 SF_1^4로의 재구성은 중복데이타의 가용성을 더욱 향상시킨다



(a) 지역이상 분포벡터 SF_3^4 = (0, 2, 2)인 경우



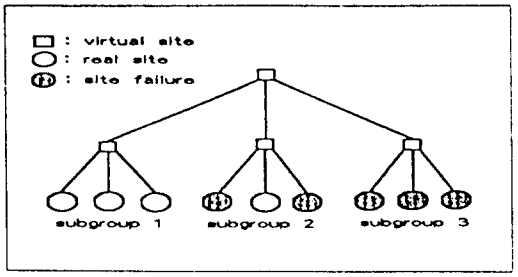
(b) HQC 구조의 재구성 후

그림 2. SF_3^4 = (0, 2, 2)인 경우 HQC 구조의 재구성

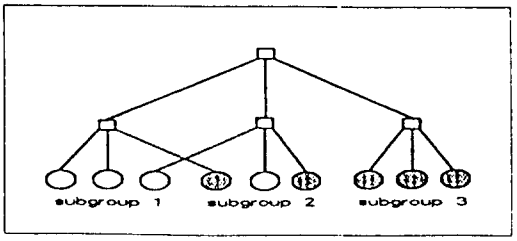
Fig. 2. Reconstruction of an HQC structure in SF_3^4=(0,2,2).

CASE 4 CASE 2의 (3) 경우

CASE 2의 (3)의 경우, 소집단 2와 3이 가용하지 않은데, 두 소집단에서 비지역이상 노드의 전체 갯수(1개)는 한 소집단내에서 동의에 필요한 최소지역수(2개)보다 작다. 따라서, CASE 3과는 달리 이 경우에는 SF_1^4를 얻기 위해서는 가용한 소집단 1의 비지역이상-노드와 소집단 2의 지역이상-노드를 교환하면 된다(그림 3)



(a) 지역이상 분포벡터 SF_3^5 = (0, 2, 3)인 경우



(b) HQC 구조의 재구성 후

그림 3. SF_3^5 = (0, 2, 3)인 경우 HQC 구조의 재구성

Fig. 3. Reconstruction of an HQC structure in SF_3^5=(0,2,3).

HQC 구조의 재구성이 CASE 3의 경우, 가용하지 않은 소집단들 사이의 노드교환에 의해 수행된 반면에, CASE 4의 경우 가용한 소집단들의 비지역이상-노드와 가용하지 않은 소집단의 지역이상 노드들의 교환에 의해 수행된다

2. HQC 구조의 재구성 알고리즘

$S = \{S_1, S_2, \cdots, S_n\}$는 중복데이타를 포함하는 n개의 지역들의 집합이라고 하고, 요구된 갱신연산의 수행여부를 결정하기 위하여 다음의 가정하에서 계층적 정족수 동의기법이 사용된다고 하자

[가 정]

(1) 집합 $S=\{s_1, s_2, \cdots, s_n\}$를 소집단 단위로 분할할 때, 각 소집단을 구성하는 노드들의 수(소집단 크기)는 동일하며, 항상 홀수개이다(이하, 특별한 언급없는 경우 소집단의 크기를 m이라 한다). 또한 S의 크기 n은 소집단 크기의 정수승이 되어야 한다 즉, $n = m^k$(단, k는 양의 정수).

(2) S의 각 노드는 지역이상(site failure)이 발생하지 않는 한 반드시 보우팅과정에 참여하며, 각 노드는 단지 하나의 소집단에만 속한다.

위의 가정을 만족하는 경우 다음과 같은 정의에 따라 계층적 정족수 동의기법에서 HQC 구조를 구성할 수 있다.

[정의 2]

집합 S가 m^k개의 지역으로서 크기가 m인 m^{k-1}개의 소집단들로 분할된다고 하자(여기서 m은 홀수, m, k는 양의 정수). 그러면 계층적 정족수 동의기법에 의해 구성되는 HQC 구조 T는 다음을 만족한다.

① T는 깊이 k+1인 m진 완전 트리 구조를 갖는다.

② 잎(leaf) 노드를 제외한 T의 각 노드들은 가상 노드이다.

③ T의 잎 노드들은 S의 원소들로서 모두 m^k개이다.

④ 레벨 i의 노드의 보우팅 값은 레벨 i+1에 있는 m개 서브트리의 루트노드의 보우팅 값들에 의해 결정된다($0 \le i \le k-1$).

HQC 구조 T의 레벨 i에 있는 노드 s 는 다음과 같이 나타낸다.

(1) i = 0일 때, s는 T의 루트노드이며 $s = \langle d_0 \rangle = \langle 1 \rangle$로 나타낸다.

(2) $i \ge 1$일 때, 노드 s의 부모노드 $s' = parent(s) = \langle d_0, \cdots, d_{i-1} \rangle$이라 하면, $s = \langle d_0, \cdots d_{i-1}, d_i \rangle$로 나타낸다(단, $0 \le j \le i$에 대해 $1 \le d_j \le m$이다). 여기서 d_i는 부모노드 s'의 서브트리에 대한 순서 인덱스이다.

예를 들어 그림 1(a)에서 소집단 3의 노드 중 가용노드는 〈1, 3, 1〉로, 그림 2(a)에서 소집단 2의 노드 중 가용노드는 〈1, 2, 2〉로 나타낼 수 있다.

한편, HQC 구조 T의 레벨 i에 있는 노드 s에 대한 가용성 함수는 다음과 같이 정의된다.

[정의 3]

레벨 i의 노드 $s = \langle d_0, \cdots, d_i \rangle$의 가용성 여부를 나타내는 함수 av(s)는 다음과 같이 정의 한다. 즉,

$$av(s) = \begin{cases} \text{'avail'} & \text{if s is available} \\ \text{'fail'} & \text{if s is not available} \end{cases}$$

이다. 여기서 레벨 i의 노드 s가 보우팅 과정에서 가용하다라는 것은 노드 s의 자식노드들 중 가용한 노드의 수가 과반수를 넘는다는 것을 의미한다. 노드 s가 잎 노드인 경우는 s가 지역이상이 있을 경우 av(s) = 'fail'이 된다. 또한, s가 HQC 구조 T의 루트노드인 경우 av(s) = 'avail'이면 T가 가용하다라고 말한다.

[정리 1]

중복데이타를 포함하는 노드들의 집합 S에 대해, 계층적 정족수 동의기법에 의해 구성된 HQC 구조 T의 깊이가 d, 차수가 m이라고 하자($d \ge 3$, m은 홀수). 이때 T가 가용하다면 적어도 $\lceil m/2 \rceil^{d-1}$개의 가용한 잎노드(비지역이상노드)들이 존재한다.

〈증명〉

i) d = 3일 때, 레벨 0의 노드 〈1〉은 가용하므로 계층적 정족수 동의기법에 따라 〈1〉의 자식노드들중 적어도 $\lceil m/2 \rceil$개는 가용하다. 인덱스를 조정하여 〈1,1〉, 〈1,2〉, …, 〈1, $\lceil m/2 \rceil$〉가 가용하다 하더라도 증명의 일반성을 잃지 않는다. 그러면 레벨 1의 가용노드 〈1,i〉 (i=1,2,…, $\lceil m/2 \rceil$) 각각에 대해 레벨 2에는 가용한 노드가 적어도 $\lceil m/2 \rceil$개 존재한다. 따라서, 레벨 2에는 총 $\lceil m/2 \rceil * \lceil m/2 \rceil = (\lceil m/2 \rceil)^2 = \lceil m/2 \rceil^{d-1}$개의 가용노드가 존재한다.

ii) d = k일 때, 위 사실이 성립한다고 가정하자. 그러면 루트노드 s = 〈1〉이 가용하다면 적어도 $\lceil m/2 \rceil^{k-1}$개의 가용노드들이 존재한다. 이제 d = k+1 이라고 하자. 그러면, 레벨 k의 가용노드들 각각에 대해 레벨 k+1에는 적어도 $\lceil m/2 \rceil$개의 가용한 자식노드들이 존재한다. 따라서, 레벨 k+1에는 적어도 $\lceil m/2 \rceil^{k-1} * \lceil m/2 \rceil = \lceil m/2 \rceil^k = \lceil m/2 \rceil^{d-1}$개의 가용노드들이 존재한다.

iii) 위의 i), ii)로부터 성립한다. ■

이 정리는 d = 1, 2인 경우도 성립하지만 계층적 정족수 동의기법의 효율성을 감안하여 소집단으로 묶여지는 경우인 $d \ge 3$으로 제한하였다.

한편, 중복데이타를 포함하는 노드들의 집합 S에 대해, 계층적 정족수 동의기법에 의해 구성된 HQC 구조 T의 깊이가 d, 차수가 m이라고 할 때, 지역이상이 발생하지 않은 노드의 갯수가 $(\lceil m/2 \rceil)^{d-1}$ 이상이면 HQC 구조의 재구성과정에 의해 T는 가용성을 갖을 수 있다. 이제 이러한 HQC 구조의 재구성과정이 어떻게 구성되는지에 대해 설명한다.

먼저 몇개의 용어를 정의한다. 임의의 레벨에 있는 노드 s의 자식 노드들의 집합을 set_child(s)로 나타

낸다. 그리고, 노드 s에 대해 하위 가용노드의 집합을 set_avail(s)라 하고 하위 비가용노드의 집합을 set_fail(s)라 한다. 즉,

$$set\_avail(s) = \{t \mid t \in set\_child(s),\ av(t) = \text{'avail'}\}$$
$$set\_fail(s) = \{t \mid t \in set\_child(s),\ av(t) = \text{'fail'}\}$$

이다. 또한 노드 s의 자식노드 중, 가용하지 않은 노드들의 집합을 N(s)이라고 한다. 즉,

$$N(s) = \{t \mid t \in set\_child(s),\ av(t) = \text{'fail'}\}$$

이다. 그리고, N(s)의 각 원소 t에 대해 t의 자식노드 중 가용 노드들의 집합을 SN(s), 비가용 노드들의 집합을 FN(s)라고 한다. 즉,

$$SN(s) = \{u \mid u \in set\_child(t),\ av(u) = \text{'avail'},\ t \in N(s)\}$$
$$FN(s) = \{u \mid u \in set\_child(t),\ av(u) = \text{'fail'},\ t \in N(s)\}$$

이다.

그러면, av(⟨1⟩) = 'fail'에 대한 HQC 구조의 재구성은 다음 소절에서 다루는 바와 같이 두 방법이 있으며 어느 방법을 선택하는가의 기준은 그림 4의 조건문과 같이 표현된다

Input: 중복데이타를 포함하는 노드들의 집합 S에 대한 깊이가 d, 차수가 m이고 가용하지 않은 HQC 구조 T

Output: av(⟨1⟩)

```
begin
  if ((card(available sites in S)) < (⌈m/2⌉)^(d-1)) then
    av(<1>) ← 'fail'
  else
    i ← 0;
    do
      do
        check_it ← 'false';
        if (card(SN(s)) ≥ ⌈m/2⌉) then
          call Reconstruction_method I ;
          check_it ← 'true'
        else
          call Reconstruction_method II ;
          check_it ← 'true'
        endif;
      while (check_it ← 'true');
      i ← i + 1;
    while ((i ≤ d-3) or (av(<1>) = 'fail'))
  endif
end;
```

그림 4. 메인 모듈

Fig. 4. Main module.

1) HQC 구조의 재구성 방법 I

레벨 i의 노드 s = ⟨$d_0,\cdots,d_i$⟩가 가용하지 못하고, card(SN(s)) ≥ ⌈m/2⌉ 이면, N(s)에 속하는 노드를 루트로 갖는 서브트리의 구조를 논리적으로 재구성하여 노드 s의 가용정족수를 1만큼 증가시킬 수 있다.

N(s)의 원소 중 하위 가용노드 집합이 가장 큰 노드(s_M)를 선택하여 set_fail(s_M)의 한 원소와 ∪ set_avail(s')의 한 원소를 서로 교환함으로써 논리적으로 재구성한다. 이 과정은 s'∈N(s)-(s_M)card (set_avail(s_M))≥ ⌈m/2⌉ 가 될 때까지 반복한다. 그러면 결국 av(s_M)= 'avail'이 되어 노드 s의 가용자식노드수가 1 만큼 증가하게 된다. 세부과정은 그림 5의 알고리즘과 같다

```
Procedure Reconstruction_method I
  /* s = <d0,....,di> : av(s) = 'fail'인 레벨 i의 노드
  /* N(s) = (ns1, ns2, ... , nsk) : 노드 s의 비가용 자식노드들의 집합
  /* MN(s) : 하위 가용노드집합이 최대가 되는 N(s)의 원소들의 집합
  /* SN(s) : N(s)의 모든 노드의 하위 가용노드들의 집합
  /* FN(s) : N(s)의 모든 노드의 하위 비가용노드들의 집합
    begin
    1. Construct the set MN(s);
      1.1 Max ← 0;
      1.2 for i = 1 to k do
            if (card(set_avail(nsi)) > Max) then
              Max ← card(set_avail(nsi));
      1.3 MN(s) ← φ;
      1.4 For each nsi ∈ N(s),
            if (card(set_avail(nsi)) = Max) then
              MN(s) ← MN(s) ∪ (nsi);
    2. Select one element sM in MN(s);
    3. Construct the sets SN(s), FN(s) for N(s)-(sM);
      3.1 SN(s) ← φ; FN(s) ← φ;
      3.2 For each ns ∈ N(s)-(sM),
          3.2.1 SN(s) ← SN(s) ∪ set_avail(ns);
          3.2.2 FN(s) ← FN(s) ∪ set_fail(ns);
    4. exchange one element in SN(s) and one element in set_fail(sM);
      repeat
        4.1 select one element a in SN(s);
        4.2 select one element b in set_fail(sM);
        4.3 exchange a and b;
            4.3.1 SN(s) ← SN(s) - (a);
            4.3.2 FN(s) ← FN(s) ∪ (b);
            4.3.3 set_fail(sM) ← set_fail(sM) - (b);
            4.3.4 set_avail(sM) ← set_avail(sM) ∪ (a);
      until (card(set_avail(sM)) ≥ ⌈m/2⌉)
    end;
```

그림 5. HQC 구조의 재구성 알고리즘 I

Fig. 5. Reconstruction algorithm I of HQC structure.

2) HQC 구조의 재구성 방법 II

레벨 i의 노드 s = ⟨$d_0,\cdots,d_i$⟩가 비가용이고 card (SN(s)) < ⌈m/2⌉ 인 경우에는 N(s)만의 논리적 재구성으로는 노드 s의 가용노드수를 증가시킬 수 없기 때문에 집합 set_avail(s)의 각 노드가 가지고 있는 여분의 가용노드들을 이용해야 한다. 즉, set_avail(s)의 어떤 노드 t 는 card(set_avail(t))> ⌈m/2⌉ 로 정족수외의 여분의 가용노드를 가질 수

있어서 이 노드들을 이용하여 set_fail(s)의 어떤 노드를 가용하게 만드는 방법이다.

여기서, 집합 SS(s)를 SS(s) = {u | u ∈ set_avail(t), t ∈ set_avail(s)} 로 정의하고 집합 AS(s)를 set_avail(t)의 각 원소가 가용성을 유지할 수 있는 최소의 가용노드들 전체(그 갯수는 card (set_avail(s)) * ⌈m/2⌉ 이다)를 SS(s)에서 뺀 SS (s)의 부분집합이라 하자. 그러면, set_fail(SM)의 한 노드와 SN(s)∪AS(s)의 한 노드를 서로 교환하여, HQC 구조를 재구성한다. 이 과정은 card(set_avail (SM)) ≥ ⌈m/2⌉ 가 될 때까지 반복한다. 결국 av(SM) = avail ' 이 되어 노드 s의 가용자식노드수가 1 만큼 증가된다. 세부과정은 그림 6의 알고리즘과 같다

```
Procedure Reconstruction_method Ⅱ
 /* AS(s) : set_avail(s)내 각 노드의 하위 가용노드들의 집합
    begin
    1. Construct the set MN(s);
    2. Select one element s_w in MN(s);
    3. Construct the sets SN(s), FN(s) for N(s)-(s_w);
    4. Construct the set AS(s) from set_avail(s);
      4.1 AS(s) ← φ;
      4.2 For each t ∈ set_avail(s),
          4.2.1 count ← 0;
          4.2.2 for each u ∈ set_avail(t),
                count ← count + 1;
                if (count > ⌈m/2⌉) then
                  AS(s) ← AS(s) ∪ {u};
    5. exchange one element in AS(s)∪SN(s) and one element in set_fail(s_w);
       repeat
         5.1 select one element v in AS(s) ∪ SN(s);
         5.2 select one element w in set_fail(s_w);
         5.3 exchange v and w;
             5.3.1 set_fail(s_w) ← set_fail(s_w) - {w};
             5.3.2 set_avail(s_w) ← set_avail(s_w) ∪ {v};
             5.3.3 AS(s) ∪ SN(s) ← AS(s) ∪ SN(s) - {v};
             5.3.4 set_fail(parent(v)) ← set_fail(parent(v)) ∪ {w};
       until (card(set_avail(s_w)) ≥ ⌈m/2⌉);
    end.
```

그림 6. HQC 구조의 재구성 알고리즘 Ⅱ

Fig. 6. Reconstruction algorithm Ⅱ of HQC structure.

Ⅴ. 가용성 분석

중복데이타의 일관성 유지를 위한 보우팅기법에서 가용성이란 지역이상 노드의 발생이 보우팅기법의 수행에 미치는 영향성을 평가하기 위한 척도로, 일반적으로 보우팅기법을 위한 가용노드의 확보가능성을 의미한다. 본 논문에서는 정족수 동의기법들의 가용성에 대한 해석적 분석을 위해 정족수 동의기법에 대한 가용성을 지역이상 노드의 발생에 따른 가용노드의 확보확률로서 정의한다. 이러한 정의에 따르면, 지역이상 노드의 발생수가 증가할수록 가용성은 감소하게 되며 지역이상 노드의 수가 특정수(가용한계점)를 넘게 되면 가용성은 0이 된다. 따라서 가용성은 다음 정의 4의 가용성 평가함수로 표현할 수 있다.

[정의 4]

중복데이타를 갖는 전체 노드의 수를 n이라고 하고 정수 a, b에 대해 a부터 b까지의 정수집합 {a,a+1, …,b}를 〈a, b〉라 하자. 그러면 정족수 동의기법에 대한 가용성 평가함수 $A:\langle 0,n\rangle \rightarrow [0,1]$ 은 단조감소함수로서 다음의 성질을 만족한다.

(1) $A(x) = 1 \ (x = 0)$

$0 \ (x = n)$

(2) 다음의 성질을 만족하는 점 p가 구간 〈0, n〉에 존재한다.

① 구간 〈0, p〉에서 A(x)는 단조감소함수이다.

② 구간 〈p, n〉에서 A(x)는 A(x) = 0 으로 상수함수이다.

여기서, 점 p를 A의 가용한계점이라고 한다.

이제 정족수 동의(QC)기법, 계층적 정족수 동의(HQC)기법, 그리고 본 논문에서 제안한 수정된 HQC(Modified HQC: MHQC)기법에 대한 가용성을 분석한다. 소집단의 크기는 m(양의 홀수)이고, 중복데이타를 가진 노드의 수는 $n=m^k$(k는 양의 정수)이라 하고 가용성 평가함수는 직선함수를 사용하기로 한다. 그리고 QC, HQC, MHQC 기법에 대한 가용성 평가함수를 각 A_{QC}, A_{HQC}, A_{MHQC}라고 하자.

QC 기법은 지역이상 노드의 수가 $\lceil m^k/2 \rceil$ (즉, $\lceil n/2 \rceil$)이상이면 가용하지 않게 된다. HQC 기법은 지역이상 노드의 수가 $m^k-(\lceil m/2 \rceil)^k$보다 크면 가용하지 않게 되고, 그 이외의 지역이상 노드의 수를 갖는 경우는 지역이상의 발생분포에 따라 가용여부가 결정된다. 그러나 제안한 MHQC 기법은 지역이상 노드의 수가 $m^k-(\lceil m/2 \rceil)^k$ 이하이면 가용성 A_{MHQC}는 $A_{MHQC}>0$으로 언제나 가용하고 $m^k-(\lceil m/2 \rceil)^k+1$ 이상이면 비로소 가용하지 않게 된다.

가용성 평가함수 A_{QC}, A_{HQC}, A_{MHQC}의 기울기를 μ_{QC}, μ_{HQC}, μ_{MHQC}라 하면 항상 다음의 관계를 갖는다.

$$-\frac{1}{\lceil m^k/2 \rceil} = \mu_{QC} \le \mu_{HQC} \le \mu_{MHQC} = \frac{1}{m^k-(\lceil m/2 \rceil)^k+1}$$

즉, HQC 기법의 가용한계점은 지역이상의 발생분포에 따라 구간 $[\lceil m^k/2 \rceil, m^k-(\lceil m/2 \rceil)^k+1]$ 사이에서 존재하게 된다.

예를 들어 m=3일 때 n=9와 27인 경우에 대한 가용성을 분석하면 다음과 같다.

n=9일 때, QC 기법은 지역이상 노드의 수가 5이상이면 가용하지 않게 되고, HQC 기법은 지역이상

노드의 수가 6 이상일 경우 가용하지 않게 되며, 4 또는 5일 경우에는 지역이상의 발생분포에 따라 가용여부가 결정된다. 그러나, MHQC 기법에서는 지역이상 노드의 수가 6이상 일 경우에만 가용하지 않게 된다. 또한, 기울기 μ_{QC}, μ_{HQC}, μ_{MHQC}의 관계는 다음과 같다.

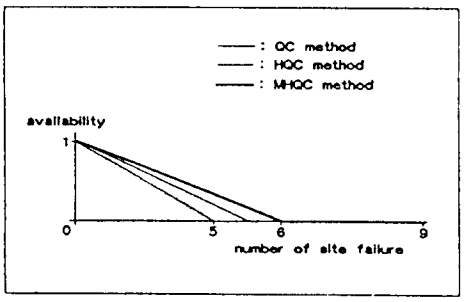
$$-\frac{1}{5} = \mu_{QC} \leq \mu_{HQC} \leq \mu_{MHQC} = -\frac{1}{6}$$

A_{HQC}의 가용한계점은 HQC 기법의 지역이상의 발생분포에 따라 구간 [5, 6] 사이에서 존재하게 된다.

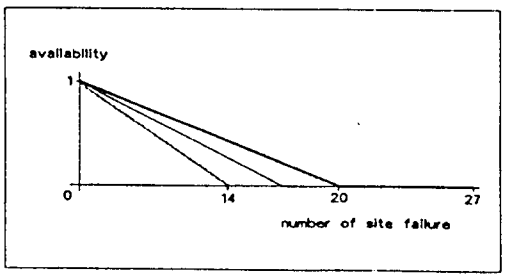
n=27일 때, QC 기법은 지역이상 노드의 수가 14이상이면 가용하지 않게 되고, HQC 기법은 지역이상 노드의 수가 20이상일 경우 가용하지 않게 되며, 그 이외에는 지역이상의 발생분포에 따라 가용여부가 결정된다. 그러나, MHQC 기법에서는 지역이상 노드의 수가 20이상일 경우에만 가용하지 않게 된다. 또한, 기울기 μ_{QC}, μ_{HQC}, μ_{MHQC}의 관계는 다음과 같다.

$$-\frac{1}{14} = \mu_{QC} \leq \mu_{HQC} \leq \mu_{MHQC} = -\frac{1}{20}$$

A_{HQC}의 가용한계점은 HQC 기법의 지역이상의 발생분포에 따라 구간 [14, 20] 사이에서 존재하게 된다.



(a) 중복데이타를 가진 지역의 수가 9인 경우



(b) 중복데이타를 가진 지역의 수가 27인 경우

그림 7. 지역이상수에 대한 가용성
Fig. 7. Availability versus number of site failure.

그림 7의 (a)와 (b)는 각각 n=9, n=27 일 때의 가용성 평가함수를 나타낸다. 그림 7에서 함수 A_{MHQC}의 가용한계점(또는 기울기)은 언제나 A_{QC}, A_{HQC}의 가용한계점(또는 기울기)보다 크거나 같다. 즉, 본 논문에서 제안한 방법은 항상 QC 기법과 HQC 기법보다 높은 가용성을 갖는다.

Ⅵ. 결론

분산시스템에서 중복데이타의 일관성을 유지하고 가용성을 높여 주는 것은 매우 중요한 일이다. 중복데이타의 일관성 유지를 위해 사용되는 계층적 정족수 동의(HQC)기법에서는 전체 지역들을 여러 개의 소집단으로 구성하여 계층적으로 보우팅과정을 수행한다. 이러한 계층적 보우팅과정은 동의에 필요한 정족수의 크기를 줄일 수 있다는 장점이 있으나, 지역이상이 발생하게 되면 지역이상의 발생분포에 따라 기존의 정족수 동의기법보다 가용성이 훨씬 떨어질 수가 있다.

따라서 본 논문에서는 데이타의 일관성 유지를 위해 계층적 정족수 동의기법을 이용할 때 중복데이타를 가지고 있는 지역에서 이상이 발생하는 경우 가용성이 떨어지는 것을 막기 위해 HQC 구조의 재구성에 의해 가용성을 향상시켜 주는 방법을 제안하였다. 또한, 기존의 정족수 동의기법, HQC 기법과 함께 본 논문에서 제안한 수정된 HQC 기법에 대해 가용성을 비교분석하였으며 그 결과 제안된 기법이 항상 정족수 동의기법이나 계층적 정족수 동의기법보다 높은 가용성을 갖는다는 것을 보여 주었다.

앞으로 계속 연구되어야 할 내용으로는 현재 계층적 정족수 동의기법에서 구성되는 계층구조가 완전트리임을 가정하고 있으므로 보다 일반화시켜 불완전 계층구조를 형성하는 경우의 HQC 기법에 관한 것이다.

參考文獻

[1] G.F. Coulouris and J. Dollimore, *Distributed Systems Concepts and Design*, Addison-Wesley Publishing Company, Inc., 1988.

[2] S.B. Davison, H. Garcia-Molina and D. Skeen, "Consistency in Patitioned Networks," *ACM Computing Surveys*, vol.17, no.3, pp.341-370, Sept 1985.

[3] S.B. Davison, "Optimism and Consistency in Patitioned Distributed Database

Systems," ACM Transactions on Database Systems, vol.9, no.3, pp.456-481, Sept. 1984.

[4] D.K. Gifford, "Weighted Voting for Replicated Data," *Proceedings of The 7th Symposium on Operating Systems Principles*, ACM, New York, pp.150-162, 1979.

[5] A. Goscinski, *Distributed Operating Systems: The Logical Design*, Addison-Wesley Publishing Company, Inc., 1991.

[6] S. Jajodia and D. Mutchler, "Dynamic Voting Algorithms for Maintaining the Consistency of a Replicated Databases," *ACM Transactions on Database Systems*, vol.15, no.2, pp.230-280, June 1990.

[7] A. Kumar, "Performance Analysis of A Hierarchical Quorum Consensus Algorithm for Replicated Objects," *The 10th Int'l Conference on Distributed Computing Systems*, pp.378-385, May 1990.

[8] A. Kumar, "Hierarchical Quorum Consensus: A New Algorithm for Managing Replicated Data," *IEEE Transactions on Computers*, vol.40, no.9, pp.996-1004, Sept. 1991.

[9] S. Mullender, Distributed Systems, Addison-Wesley Publishing Company Inc., 1989.

[10] R.H. Thomas, "A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases," *ACM Transactions on Database Systems*, vol. 4, no.2, pp.180-209, June 1979.

著 者 紹 介

劉 憲 昌(正會員)

1989年 고려대학교 전산과학과(이학사). 1991年 고려대학교 대학원 전산과학과(이학석사). 1993年 고려대학교 대학원 전산과학과 박사과정 수료. 1991年 ~ 현재 고려대학교 전산과학과 강사. 주관심 분야는 Distributed System, Distributed Operating System, Distributed Database, Computer Network 등임.

•

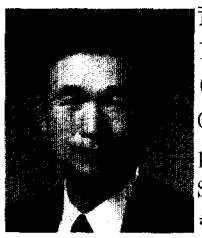
孫 進 坤(正會員)

1984年 고려대학교 수학과 졸업(이학사). 1988年 고려대학교 대학원 전산학전공(이학석사). 1991年 고려대학교 대학원 전산학전공(이학박사). 1991年 ~ 현재 한국방송통신대학교 전자계산학과 조교수. 주관심 분야는 Dis-tributed System, computer Network, Modeling and Simulation 등임.

趙 東 榮(正會員)

1986年 고려대학교 수학교육학과 졸업(이학사). 1988年 고려대학교 대학원 전산학전공(이학석사). 1992年 고려대학교 대학원 전산학전공(이학박사). 1993年 ~ 현재 전주대학교 전자계산학과 전임강사. 주관심 분야는 Distributed System, Distributed Database, Knowledge Base 등임.

•



黃 鍾 善(正會員)

1966年 고려대학교 수학과 졸업(이학사). 1978年 Univ. of Georgia, Statistics & Computer Science 박사. 1978年 South Carolina Lander 주립대학 조교수. 1982年 ~ 1990年 고려대학교 전자계산소 부소장. 1986年 ~ 1989年 한국정보과학회 부회장. 1982年 ~ 현재 고려대학교 전산과학과 교수. 주관심 분야는 Distributed System, Distributed Database.