제 18 회 한국정보처리학회 추계학술발표대회 논문집 제 9 권 제 2 호 (2002.11)

효율적이고 안정한 분산시스템을 위한 하이브리드 복제 프로토콜

최성춘*, 윤희용*, 이보경**, 최중섭**, 이형수***
*성균관대학교 정보통신 공학부
**한국정보보호진흥원 정보보호기술팀
***전자부품연구원 정보시스템연구센터
*{choisc, youn}@ece.skku.ac.kr
**{bklee, jschoi, hgkim}@kisa.or.kr, ***hslee@keti.re.kr

A Hybrid Replication Protocol for Efficient and Secure Distributed System

Sungchune Choi*, Hee Yong Youn*
Bo Kyoung Lee**, Joong Sup Choi**, Lee Hyung Su***
*School of Information and Communication, Sungkyunkwan University
**Information Security Technology Division, Korea Information Security Agency
***IT System Research Center, Korea Electronics Technology Institute

요 약

최근 분산 컴퓨팅 환경에서 데이터와 서비스의 복제는 통신 비용의 감소, 데이터 가용성 증가, 그리고 단일 서버의 병목현상을 피하기 위해 필수적이다. 기존의 대표적인 복제 프로토콜로 네트워크를 논리적으로 구성하는 Tree quorum 프로토콜과 Grid 프로토콜이 있다. Tree quorum 프로토콜은 최선의 경우 가장 우수한 읽기 성능을 보이는 반면 트리의 높이가 증가할수록 노드의 수가 기하급수적으로 증가한다는 단점을 가지고 있다. Grid 프로토콜은 읽기 동작에 있어 높은 가용성을 가지는 반면 고장이 없는 환경에서도 같은 읽기 및 쓰기 성능을 보이는 단점을 가지고 있다. 따라서 본 논문에서는 기존의 복제 프로토콜이 가지는 문제점을 해결하고, 복제 노드의 추가와 삭제가 보다 용이한 하이브리드 복제 프로토콜을 제안한다. 제안된 복제 프로토콜은 같은 수의 노드를 갖는 tree quorum 프로콜에 비해 적은 읽기 비용을 가지며, 효율적인 노드의 구성을 통해 기존 복제 프로토콜보다 높은 데이터의 가용성을 가진다.

1. 서론

오늘날 대용량 분산 컴퓨팅 환경에서 데이터와 서비스의 복제는 데이터의 가용성을 높이고, 전체 시스템의 성능을 향상시키기 위해 필수적인 기술이다 [1]. 또한 다중 노드를 사용함으로써 단일 서버의 사용으로 인한 병목현상 문제를 해결할 수 있다. 그러나 노드의 수가 증가할수록 통신 비용은 증가하게 되므로, 전체 시스템을 구성하는 노드 중에 읽기/쓰기 동작을 위하여 접속해야하는 노드의 수는 가능하면 적은 수로 유지되어야 한다[2][3].

Tree quorum 프로토콜 [4]과 grid 프로토콜 [5]은 노드들의 논리적 구성을 통하여 전체 노드중 일부의 노드만을 이용해 읽기/쓰기 동작을 수행하는 복제 프로토콜이다. 물론 일부 노드만을 사용함으로써 발생하는 일관성 문제를 해결하기 위한 일관성 제어 기법이 필요하다 [6].

Tree quorum 프로토콜은 루트 노드를 이용하여 적은 읽기 비용을 허락하는 반면 트리의 높이가 증가할수록 노드의 수가 기하급수적으로 증가한다는 단점을 가지고 있다. Grid 프로토콜은 tree quorum 프로토콜에 비해 높은 가용성을 보이는 반면 노드의 고장이 없는 환경에서도 항상 같은 읽기와 쓰기 비용을 갖는다는 단점을 가지고 있다. 또한 두가지 프로토콜 모두 효율적인 노드의 추가와 삭제가 가능하지 못하다는 단점을 가지고 있다.

따라서 본 논문에서는 기존의 tree quorum 프로토콜과 grid 프로토콜이 가지는 장점을 모두 가지면서 기존 프로토콜들의 단점을 해결할 수 있는 하이브리드 복제 프로토콜을 제안한다. 하이브리드 복제 프로토콜은 트리 네트워크와 그리드 네트워크를 혼합한 방식으로 네트워크를 구성하기 위해 세가지의 변수를 사용하여 노드의 동적 구성을 가능하도록 해준다. 이러한 네트워크 구조의 사용과 노드의 동적 구성을 통하여 기존 프로토콜에 비해 적은 동작 비용과 높은

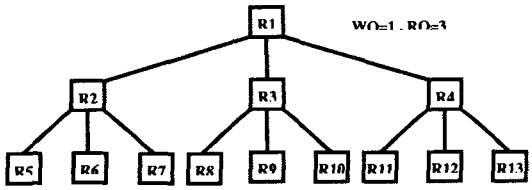
가용성을 가진다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 복제 프로토콜에 대하여 설명하고, 3 장에서는 본 논문에서 제안하는 동적인 하이브리드 복제 프로토콜을 소개하고, 기존 프로토콜과의 성능을 비교한다. 마지막으로 4장에서 논문의 결론 및 향후 과제를 제시한다.

2. 기존 연구

2.1 Tree quorum 프로토콜

Tree quorum 프로토콜은 (그림 1)과 같은 논리 트리 구성을 이용한다. 높이 h의 트리로 구성된 n개의 노드가 있다고 가정을 하면, 잎 노드를 제외한 각 노드 R_i는 S_{R_i} 수 만큼의 자식 노드를 갖는다. 각 노드들을 위한 읽기와 쓰기 집합을 정의하기 위해 read quorum rq_{R_i}와 write qrorum wq_{R_i}를 정의한다. Tree quorum 프로토콜은 다양한 rq_{R_i}와 wq_{R_i} 값을 가질수 있으며, 그에 따라 다른 성능을 보인다. 이때 rq_{R_i}의 값이 S_{R_i}의 값과 동일하고, wq_{R_i} 값이 1인 프로토콜을 Logarithmic 프로토콜이라 하며, tree quorum 프로토콜 에서 가장 좋은 성능을 나타낸다.

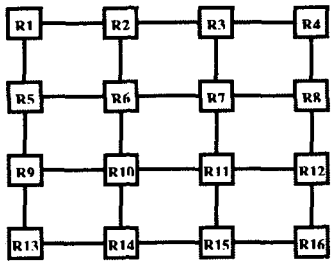


(그림 1) 13개의 노드를 갖는 높이 3의 트리

rq_{R_i} 값이 3이고 wq_{R_i} 값이 1일 경우, 읽기 동작을 위해 필요한 노드들의 집합은 {R1}, {R2, R3, R4}, {R3, R4, R5, R6, R7}, 그리고 {R3, R5, R6, R7, R11, R12, R13}이 된다. 쓰기 동작을 위해 필요한 노드들의 집합은 {R1, R2, R6}, {R1, R3, R8}, 그리고 {R1, R4, R11}이 된다.

2.2 Grid 프로토콜

Grid 프로토콜은 (그림 2)와 같이 행과 열의 논리적인 그리드 형태로 구성된다.



(그림 2) 16개의 노드를 갖는 그리드 네트워크

읽기 동작을 위해 필요한 노드들의 집합은 {R1, R6, R3, R12}와 {R5, R10, R7, R8}이 된다. 쓰기 동작을 위해 필요한 노드들의 집합은 {R1, R6, R3, R12, R2, R10, R14}와 {R1, R2, R11,R8, R4, R12, R16}이 된다.

3. 제안된 프로토콜

이전 장에서 설명한 것과 같이 tree quorum 프로토콜과 grid 프로토콜은 몇가지 문제점들을 가지고 있다. 따라서 본 논문에서는 루트 노드의 고장이 없을 경우 우수한 읽기 성능을 가지는 tree quorum 프로토콜의 장점과 높은 가용성을 가지는 grid 프로토콜의 장점을 이용한 새로운 하이브리드 복제 프로토콜을 제안한다. 제한된 프로토콜은 노드의 추가와 삭제가 용이하도록 세가지의 구성 변수를 이용하여 동적구성이 가능하다.

3.1 하이브리드 프로토콜

대부분의 복제 프로토콜과 동일하게, 본 논문에서 제안한 프로토콜에서는 노드의 고장은 발생 할 수 있지만 비잔틴 고장은 고려하지 않는다.

제안된 프로토콜은 (표1)과 같이 3개의 변수를 이용하여 노드를 구성하고, 이 변수의 결정에 따라 (그림 3)과 같이 노드들이 구성된다. 전체 네트워크를 구성하는 노드의 수는 다음과 같이 계산된다.

$$\text{노드의 수} = \left(\sum_{k=0}^{h-1} s^k\right) + \left(s^{h-1} * g\right)$$

따라서 같은 수의 노드를 이용하여 다른 형태의 네트워크 구성이 가능하며 노드의 추가/삭제가 용이하다.

(표1) 노드를 구성하기 위한 변수

변수 (h, s, g)	정의
h	트리의 높이
s	자식의 수
g	그리드 네트워크의 깊이

(그림 3)에서 보는것과 같이, 제안된 프로토콜의 동작을 설명하기 위한 알고리즘은 높이 h의 트리 구조 부분과 깊이 g의 그리드 구조 부분으로 나뉘어진다. 또한 그리드 구조는 트리의 자식수에 따라 그룹으로 나뉘어 독립적으로 동작하게 된다.

- *읽기 동작*

트리 구조에서의 읽기 동작은 루트 노드로부터 시작하고, 만약 루트노드가 실패할 경우에는 자식 노드 전체에 대하여 읽기를 수행한다. 루트의 자식 노드는 다시 루트와 같이 동작하게 되며, 트리의 끝에 도달할때까지 반복하여 동작한다.

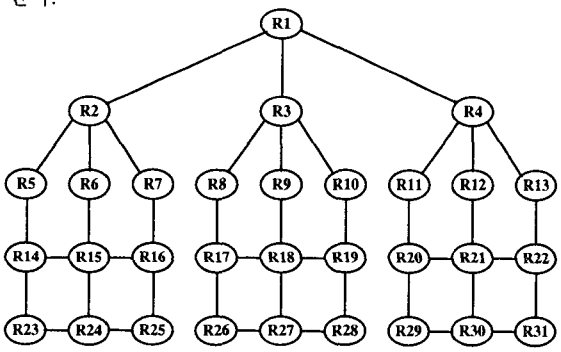
그리드 구조에서의 읽기 동작은 각 열의 전체 노드들에 대하여 동작하게 되고, 만일 각 열에서 하나의 노드라도 실패할 경우에는 다음 열에 대하여 같은 동작을 반복하여 수행하게 된다.

- *쓰기 동작*

트리 구조에서의 쓰기 동작은 읽기 동작과 마찬가지로 루트 노드로부터 시작하고, 루트의 자식 노드 중에 임의의 하나의 노드를 선택하여 쓰기 동작을 수

행한다. 선택된 루트의 자식 노드는 루트와 같이 동작하게 되고, 트리의 끝까지 반복하여 동작하게 된다.

그리드 구조에서의 쓰기 동작은 각 열의 노드들 중에 임의의 하나의 노드에 대하여 쓰기 동작을 수행한다.



(그림 3) (3, 3, 2)의 변수를 이용한 노드의 구성

(그림 3)에서 읽기 동작을 위한 가능한 노드들의 집합은 {R1}, {R2, R3, R4}, {R3, R4, R5, R6, R7}, 그리고 {R3, R4, R14, R15, R16}가 된다. 쓰기 동작을 위한 가능한 노드들의 집합은 {R1, R2, R5, R15, R24}, {R1, R3, R9, R17, R26}, 그리고 {R1, R4, R12, R22, R30}이 된다.

일관성 유지를 위해 읽기 집합와 쓰기 집합은 반드시 하나 이상의 노드가 중복되어야 하며, 읽기/쓰기 동작과 쓰기/쓰기 동작은 동시에 발생하지 않도록 해야 한다. 제안된 프로토콜은 읽기 동작과 쓰기 동작 모두 루트노드를 시작으로 하기 때문에 읽기/쓰기와 쓰기/쓰기 동작이 동시에 발행할 수없다. 또한 읽기 동작은 트리는 각 레벨 전체를 읽고 그리드 구조에서는 각 열의 전체 노드를 읽기 때문에 읽기 동작이 수행되고 있는 동안에는 쓰기 동작이 수행될 수 없다.

제안된 프로토콜의 단점은 모든 읽기/쓰기 동작이 루트로부터 시작되기 때문에 루트 노드의 병목현상이 발생될 수 있다는 것이다. 이러한 단점은 읽기 동작을 수행할 때 트리구조 또는 그리드 구조에서 임의의 레벨에서 시작할 수 있도록 변경해줌으로써 해결될 수 있다. 실제로 읽기 동작은 트리에서 임의의 레벨과 그리드 구조에서 임의의 열의 전체 노드에 대하여 동작하게 되고, 쓰기는 각 레벨에서 하나의 노드만을 필요로 하기 때문에 두가지 중에 한가지를 위에 제안한 방식으로 변경해 주어도 문제가 발생하지 않는다.

3.2 성능평가

제안된 프로콜은 변수 h, s, 그리고 g에 따라 다양한 노드의 구성이 가능하고, 그에 따라 다양한 성능을 보인다. 따라서 이번 장에서는 변수의 변화에 따라 가장 우수한 성능을 보이는 구조에 대하여 제시하려 한다. 정확한 성능평가를 위하여 다음과 같은 전제를 사용한다.

- 링크의 실패는 발생하지 않는다.
- 노드들의 실패는 독립적이며 실패율은 일정하다.
- 트리 구조에서 잎 노드를 제외한 각 노드들은 같은 수의 자식을 갖는다.

3.2.1 비용 분석

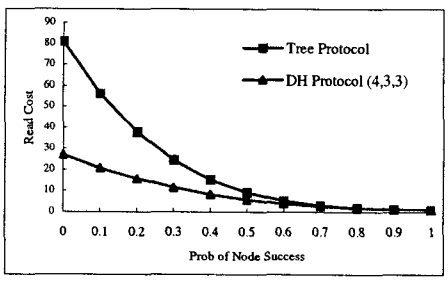
비용은 전체 노드들중에 읽기와 쓰기 동작을 수행하는 노드의 수를 의미한다.

제안된 프로토콜의 읽기 비용은 다음과 같이 계산된다.

$$C_{read} = \sum_{k=0}^{h-1}\left((1-f)^k \cdot f \cdot RQ^k\right) + (1-f)^{h-1}\sum_{k=1}^{g-1}\left(\left(1-f^s\right)^k \cdot f \cdot RQ^{h-1}\right) + \left(1-\left(\sum_{k=0}^{h-1}(1-f)^k \cdot f + (1-f)^{h-1}\sum_{k=1}^{g-1}\left((1-f^s)^k \cdot f\right)\right)\right) \cdot RQ^{h-1}$$

위의 수식에서 보는것과 같이, 평균 읽기 비용은 h와 s의 값에 따라 변화하게 된다. 이 수식에서 중요한 점은 g(그리드 네트워크의 깊이)가 증가하여도 평균 읽기 비용은 변화하지 않는다는 것이다. 읽기 비용은 깊이에 의존하지 않고 행의 수에 의존한다는 그리드 구조의 특성 때문이다. 그러므로 만약 관리자가 노드의 수를 증가시키면서 동일한 읽기 비용을 원한다면, 트리의 구조에 변화를 주지 않으면서 그리드 구조의 깊이만을 증가시키면 된다.

제안된 프로토콜을 위한 쓰기 비용은 각 레벨에서 하나의 노드에만 쓰기를 수행하기 때문에 트리의 깊이와 그리드 구조의 깊이에 의존하게 된다. (그림 4)는 121개의 노드를 갖는 tree quorum프로토콜과의 읽기 비용을 비교한 것이다.



(그림 4) 121개의 노드에서 읽기 비용 비교

(그림 4)에서 보는것과 같이 같은 수의 노드를 가질 때 평균 읽기 비용은 제안된 프로토콜이 Tree quorum 프로토콜에 비해 훨씬 우수한 것으로 나타난다. 이러한 결과를 보이는 이유는, 121개의 노드를 가지기 위해서는 트리의 경우 전체 잎 노드의 수가 81개가 되지만, 제안된 프로토콜의 경우는 27개 되기 때문이다. 또한 위에서 말한것과 같이 제안된 프로토콜의 구조에서 그리드의 깊이를 한 단계 증가하면 노드의 수는 27개 증가하지만 읽기 비용에는 변화가 없다는 것이다.

3.2.2 가용성 분석

여기에서는 제안된 프로토콜의 가용성을 평가하려한다. 제안된 프로토콜의 전체 가용성을 분석하기 위해서는 트리 구조의 가용성과 그리드 구조의 가용성으로 나누어서 분석해야 한다.

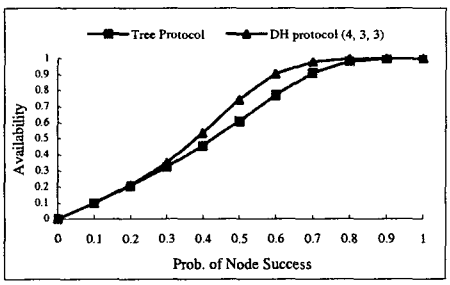
제안된 프로토콜의 그리드 구조를 위한 읽기 가

용성은 다음과 같이 순환적 공식에 의하여 구할 수 있으며, 여기서 $\wp_{read}^{G(0)} = p$ 이다.

$$\wp_{read}^{G(g)} = p^s + (1-p^s)\cdot\wp_{read}^{G(g-1)}$$

전체 읽기 가용성은 다음과 같이 구할수 있으며, 여기서 $\wp_{read}^{(l)} = p + (1-p)\left(\wp_{read}^{(0)}\right)$, $\wp_{read}^{(0)} = \wp_{read}^{G(0)}$, 그리고 $l = h-1$ 이다.

$$\wp_{read}^{(l)} = p + (1-p)\left(\wp_{read}^{(l-1)}\right)^s$$



(그림 5) 읽기 가용성 비교

(그림 5)는 제안된 프로토콜과 tree quorum프로토콜의 가용성을 비교한 결과이다. 제안된 프로콜의 읽기 가용성은 tree quorum프로토콜에 비해 좋게 나타난다. 그 이유는 이전 장에서 설명한것과 같이 그리드 구조의 가용성은 트리 구조에 비해 높기 때문에 트리 구조보다 더 높은 가용성을 가질수 있게 된다.

다양한 h, s, 그리고 g값에 따라 읽기 가용성을 평가한 결과, 제안된 프로토콜의 읽기 가용성은 트리의 높이와 자식의 수를 적게 할수록, 그리드 구조의 깊이를 증가 시킬수록 높게 된다. 이러한 결과는 위에 설명한 이유에 의하여 쉽게 예측될 수 있다.

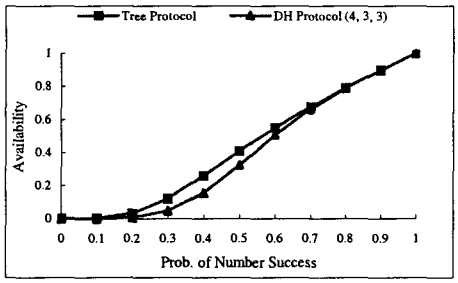
제안된 프로토콜의 쓰기 가용성을 예측하기 위해 그리드 부분의 쓰기 가용성을 구하면 다음과 같으며, 여기서 $\wp_{write}^{G(0)} = \sum_{k=1}^{s}\binom{k}{s}p^k(1-p)^{s-k}$ 그리고 $l = g-1$ 을 나타낸다.

$$\wp_{write}^{G(l)} = \left(\sum_{k=1}^{s}\binom{k}{s}p^k(1-p)^{s-k}\right)\cdot\wp_{write}^{G(l-1)}$$

전체 가용성은 다음과 같이 나타나며, 여기서 $\wp_{write}^{(0)} = p\left(\sum_{k=1}^{s}\binom{k}{s}p^k(1-p)^{s-k}\right)\cdot\wp_{write}^{G(l)}$ 그리고 $l = h-2$ 을 나타낸다.

$$\wp_{write}^{(l)} = p\left(\sum_{k=1}^{s}\binom{k}{s}\left(\wp_{write}^{(l-1)}\right)^k\left(1-\wp_{write}^{(l-1)}\right)^{s-k}\right)$$

(그림 6)은 제안된 프로토콜과 tree quorum프로토콜의 쓰기 가용성을 비교한 결과이다. (그림6)에서 트리 구조에 비해 제안된 프로토콜의 쓰기 가용성이 낮은 이유는, 트리 구조는 레벨이 증가 할수록 많은 수의 노드들을 가지게 되어 쓰기 가용성이 증가하는 반면, 제안된 프로토콜은 고정된 노드의 수로 증가하기 때문에 전체적인 쓰기 가용성이 증가되지 않는다. 하지만 세가지의 변수중에 s의 값을 증가 시켜 노드를 구성한다면 트리에 비해 훨씬 우수한 쓰기 가용성을 얻을 수 있다. 하지만 자식의 수를 증가시킬수록 읽기 비용은 상대적으로 증가하게 된다.



(그림 6) 쓰기 가용성 비교

4. 결론 및 향후 계획

본 논문은 tree quorum프로토콜과 grid프로토콜과 복제 프로토콜에 비해 우수한 성능을 보이며, 노드의 확장이 유연한 하이브리드 복제 프로토콜을 제안한다. 또한 h, s, 그리고 g의 세가지의 변수를 이용하여 최적의 성능을 갖는 노드 구성을 가능하도록 한다. 이는 서바이버블 스토리지 시스템을 위해 효율적으로 사용될 수 있는 프로토콜이다.

향후 제안된 프로토콜의 보다 정확한 성능 평가를 수행하기 위해 시뮬레이션을 수행하고, 이를 통해 응답시간과 처리율을 비교할 계획이다.

참고문헌

[1] C. Amza,, A. L. Cox, W. Zwaenepoel, Data replication strategies for fault tolerance and availability on commodity clusters, *Proceedings International Conference on Dependable Systems and Networks (DSN)*, 2000, 459-467.

[2] H.Y. Youn, D. Lee, B. K. Lee, J. S. Choi, and H. G. Kim, An Efficient Hybrid Replication Protocol for Highly Available Distributed System, *Proceedings IASTED on Communications and Computer Networks (CCN)*, Nov, 2002.

[3] D. Saha, S. Rangarajan, S. K. Tripathi, An Analysis of the Average Message Overhead in Replica Control Protocols, *Proceedings IEEE Transactions on Parallel and Distributed Systems*, 7(10), Oct, 1996, 1026-1034.

[4] D. Agrawal and A. El Abbadi, The tree Quorum protocol: An Efficient Approach for Managing Replicated Data, *Proceeding 16th Very Large Databases (VLDB) Conference*, 1990, 243-254.

[5] S. Cheung, , M. Ammar, , and M. Ahamad, , The Grid Protocol: A High Performance Scheme for Maintaining Replicated Data, *Proceedings 6th International Conference on Data Engineering*, 1990, 438-445.

[6] T. Anderson, Y. Breitbart, H. Korth, A. Wool, Replication, Consistency, and Practicality: Are These Mutually Exclusive?, *Proceedings ACM SIGMOD International Conference on Management of Data*, Jun 1998, 484-495.