논문 17-42-02-08 The Journal of Korean Institute of Communications and Information Sciences '17-02 Vol.42 No.02
https://doi.org/10.7840/kics.2017.42.2.345

이중 노드 복구가 가능한 분산 저장 시스템의 MTTDL

길 용 성◆, 김 상 효°, 박 호 성*

MTTDL for Distributed Storage Systems with Dual Node Repair Capability

Yong Sung Kil◆, Sang-Hyo Kim°, Hosung Park*

요 약

본 논문은 분산 저장 시스템의 내구도 평가 지표인 기대 데이터 수명(Mean Time To Data Loss; MTTDL)을 이중 노드 복구가 가능한 상황에서 분석하고 단일 노드 복구 상황과 비교하였다.

Key Words : distributed storage systems, reliability analysis

ABSTRACT

MTTDL, a measure for reliability of distributed storage system, is analyzed for the case when double node repair is possible and compared with the single node repair cases.

Ⅰ. 서 론

분산 저장 시스템은 많은 수의 저장 노드들이 연결되어 데이터를 저장한다. 저장 노드의 물리적 고장은 불가피하며 이들을 효율적으로 복구하기 위해 재생부호, 부분 접속 복구 부호, 피라미드 부호, 부분 반복 부호와 같은 소실 부호가 연구되었다[1,2]. 이러한 소실 부호의 성능을 평가할 수 있는 지표로 복구 대역폭, 노드 저장 용량, 최소거리, 부분 접속 수, 가용도 등이 있다[1].

한편, 실제 시스템에 소실 부호가 사용되었을 경우 시스템의 기대 수명과 같은 내구도 평가치 또한 중요한 지표가 된다. 논문 [3] 에서는 시스템 내구도를 기대 데이터 수명(Mean Time To Data Loss; MTTDL)을 통해 분석하였으며 이때 다양한 시스템 환경 요소가 고려되었다.

기존 연구들은 MTTDL 분석 시 단일 노드 복구가 가능한 시스템을 가정하였다[2-4]. 논문 [5]에서는 단일 노드 복구보다 복수 노드 복구가 복구를 위한 통신 데이터 측면에서 효율적임을 보였다. 본 논문에서는 최대 2개의 노드 복구가 가능한 시스템의 MTTDL을 분석하고 단일 노드 복구 시스템과의 비교를 통해 데이터 수명이 증가함을 보인다.

Ⅱ. 기대 데이터 수명(MTTDL) 분석

부호율 k/n과 최소거리 d를 갖는 (n,k,d) 소실 부호가 사용된 분산 저장 시스템의 상태는 그림 1의 마르코프 연쇄로 모델링 할 수 있다.

그림 1의 A는 활동(active) 중인 시스템 상태이며 다른 상태로 천이가 가능하고, F는 고장(failed)난 상태로 천이가 불가능하다. 아래 첨자는 시스템의 정상 노드 수다. 부분 접속 복구의 경우 d개 이상의 노드 혹은 부호어 심벌 소실이 발생하면 복구 불가능한 노드 혹은 부호어가 발생한다. 본 논문에서는 t_S를 MTTDL로 정의하고 그 하한을 계산하기 위해 $n-d$ 번째 상태를 F로 설정하였다.

시스템의 한 시대(epoch)를 상태 A_n에서 시작하여 A_n으로 처음 되돌아오거나 F_{n-d}에 도달하는 과정으로 정의한다. 시스템은 다수의 시대를 거치며 어떤 시대에 F_{n-d}에 도달하면 그 시대는 마지막 시대이며 시스템은 복구 불가 상태가 되고 수명을 다한다. 이때 t_S는 기대 시간 n_E와 기대 시대의 수 t_E의 곱으로 계산된다.

Q_i를 i번째 상태에서 시작하여 A_n을 거치지 않고 F_{n-d}에 도달할 확률로 정의하면 식 (1), (2), (3), (4),

※ 본 연구는 한국연구재단 글로벌박사양성사업(NRF-2016H1A2A1908244), 기본연구지원사업(NRF-2015R1D1A1A01058975)의 지원으로 수행되었음.

◆ First Author : College of Information and Communication Engineering, Sungkyunkwan University, gys0730@skku.edu, 학생회원
° Corresponding Author : College of Information and Communication Engineering, Sungkyunkwan University, iamshkim@skku.edu, 종신회원
* School of Electronics and Computer Engineering, Chonnam National University, hpark1@jnu.ac.kr, 종신회원
논문번호 : KICS2016-12-398, Received December 20, 2016; Revised January 16, 2017; Accepted January 16, 2017

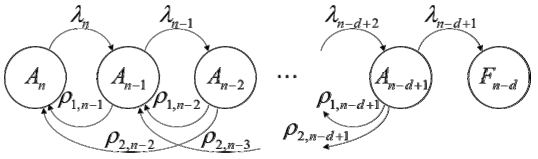

그림 1. (n,k,d) 소실 부호가 사용된 분산 저장 시스템의 마르코프 연쇄 모델
Fig. 1. Markov chain model for distributed storage system with (n,k,d) erasure code

(5)로 표현된다. 이때 파라미터 $\lambda_i, \rho_{1,i}, \rho_{2,i}$는 각각 상태 i의 단일 노드 고장률, 단일 노드 복구율, 이중 노드 복구율이다.

$$Q_i = \begin{cases} 1, & i = n-d \\ p_i Q_{i-1} + q_{1,i} Q_{i+1} + q_{2,i} Q_{i+2}, & i \in [n-d+1, n-2] \\ p_{n-1} Q_{n-2}, & i = n-1 \\ 0, & i = n \end{cases} \tag{1}$$

$$Q_{n-1} = \frac{1}{n_E} \tag{2}$$

$$p_i = \begin{cases} \dfrac{\lambda_i}{\lambda_i + \rho_{1,i} + \rho_{2,i}}, & i \in [n-d+1, n-2] \\ \dfrac{\lambda_{n-1}}{\lambda_{n-1} + \rho_{1,n-1}}, & i = n-1 \\ 1, & i = n \end{cases} \tag{3}$$

$$q_{1,i} = \begin{cases} \dfrac{\rho_{1,i}}{\lambda_i + \rho_{1,i} + \rho_{2,i}}, & i \in [n-d+1, n-2] \\ \dfrac{\rho_{1,n-1}}{\lambda_{n-1} + \rho_{1,n-1}}, & i = n-1 \end{cases} \tag{4}$$

$$q_{2,i} = \frac{\rho_{2,i}}{\lambda_i + \rho_{1,i} + \rho_{2,i}}, \; i \in [n-d+1, n-2] \tag{5}$$

R_i를 식 $R_i = Q_{i-1} - Q_i$로 정의하면 R_i는 식 (6), (7), (8)과 같이 표현된다.

$$R_i = \begin{cases} a_i R_{i+1} + b_i R_{i+2}, & i \in [n-d+1, n-2] \\ a_{n-1} Q_{n-1}, & i = n-1 \\ Q_{n-1}, & i = n \end{cases} \tag{6}$$

$$a_i = \begin{cases} \dfrac{q_{1,i} + q_{2,i}}{p_i}, & i \in [n-d+1, n-2] \\ \dfrac{q_{1,n-1}}{p_{n-1}}, & i = n-1 \end{cases} \tag{7}$$

$$b_i = \begin{cases} \dfrac{q_{2,i}}{p_i}, & i \in [n-d+1, n-2] \\ 0, & i = n-1 \end{cases} \tag{8}$$

이제 R_i는 c_i를 통해 다음과 같이 표현된다.

$$c_i = \begin{cases} 0, & i = n-d+1 \\ \dfrac{b_{i-1}}{c_{i-1} - a_{i-1}}, & i \in [n-d+2, n-1] \end{cases} \tag{9}$$

$$R_i = c_i R_{i+1} + d_i Q_{n-1} \tag{10}$$

$$d_i = \prod_{j=i}^{n-1} (a_j - c_j), \; i \in [n-d+1, n-1] \tag{11}$$

식 (10)을 전개하고 R_i의 합을 계산하면 식 (12), (13), (14)를 얻을 수 있으며 n_E는 식 (15)와 같다.

$$R_i = e_i Q_{n-1} \tag{12}$$

$$e_i = \prod_{j=i}^{n-1} c_j + \sum_{j=i}^{n-1} \left(\prod_{l=i}^{j-1} c_l \right) d_j, \; i \in [n-d+1, n] \tag{13}$$

$$\sum_{j=n-d+1}^{n} R_j = \sum_{j=n-d+1}^{n} e_j Q_{n-1} = 1 - 0 \tag{14}$$

$$n_E = \sum_{j=n-d+1}^{n} e_j \tag{15}$$

T_i를 i번째 상태에서 시작하여 A_n이나 F_{n-d}에 도달할 시간으로 정의하고 t_i를 i번째 상태에 머무르는 시간으로 정의하면 식 (16), (17)과 같다.

$$T_i = \begin{cases} 0, & i = n-d \\ p_i T_{i-1} + q_{1,i} T_{i+1} + q_{2,i} T_{i+2} + t_i, & i \in [n-d+1, n-2] \\ p_{n-1} T_{n-2} + q_{1,n-1} T_n + t_{n-1}, & i = n-1 \\ 0, & i = n \end{cases} \tag{16}$$

$$t_i = \begin{cases} \dfrac{1}{\lambda_i + \rho_{1,i} + \rho_{2,i}}, & i \in [n-d+1, n-2] \\ \dfrac{1}{\lambda_{n-1} + \rho_{1,n-1}}, & i = n-1 \\ \dfrac{1}{\lambda_n}, & i = n \end{cases} \tag{17}$$

U_i를 식 $U_i = T_{i-1} - T_i$로 정의하면 식 (18)과 같고 c_i를 이용하면 식 (19), (20)과 같이 표현된다.

$$U_i = \begin{cases} a_i U_{i+1} + b_i U_{i+2} - \dfrac{t_i}{p_i}, \\ \qquad i \in [n-d+1, n-2] \\ a_{n-1} T_{n-1} - \dfrac{t_{n-1}}{p_{n-1}}, i = n-1 \\ T_{n-1}, \qquad i = n \end{cases} \tag{18}$$

$$U_i = c_i U_{i+1} + d_i T_{n-1} - f_i \tag{19}$$

$$f_i = \sum_{j=i}^{n-1} \left(\prod_{l=i}^{j-1} (a_l - c_l) \right) \frac{t_j}{p_j}, i \in [n-d+1, n-1] \tag{20}$$

식 (19)를 전개하고 U_i의 합을 계산하면 식 (21), (22), (23)를 얻을 수 있다. 따라서 t_E와 t_S는 식 (24), (25)와 같다.

$$U_i = e_i T_{n-1} - g_i \tag{21}$$

$$g_i = \sum_{j=i}^{n-1} \left(\prod_{l=i}^{j-1} c_l \right) f_j, \; i \in [n-d+1, n] \tag{22}$$

$$\sum_{j=n-d+1}^{n} U_j = \underbrace{\sum_{j=n-d+1}^{n} e_j T_{n-1} - \sum_{j=n-d+1}^{n} g_j}_{=0} \tag{23}$$

$$\begin{aligned} t_E &= t_n + T_{n-1} \\ &= \frac{1}{\lambda_n} + \left(\sum_{j=n-d+1}^{n} e_j \right)^{-1} \sum_{j=n-d+1}^{n} g_j \end{aligned} \tag{24}$$

$$\begin{aligned} t_S &= n_E t_E \\ &= \frac{1}{\lambda_n} \sum_{j=n-d+1}^{n} e_j + \sum_{j=n-d+1}^{n} g_j \end{aligned} \tag{25}$$

Ⅲ. 분산 저장 시스템의 MTTDL 성능 분석

이중 노드 복구가 가능한 시스템의 최대, 최소 t_S 비교를 통해 단일 노드 복구와 이중 노드 복구의 t_S 성능을 비교한다. 이를 위해 W_i를 총 노드 복구율 대비 단일 노드 복구율 $W_i = \rho_{1,i} / (\rho_{1,i} + \rho_{2,i})$, $W_{n-1} = 1$, ψ를 단일 노드 복구 대비 이중 노드 복구의 t_S의 비 $\psi = t_{S,W_i=0} / t_{S,W_i=1}$로 정의한다.

t_S 분석을 위한 시스템 모델은 [4]를 따르며 $d=6$,

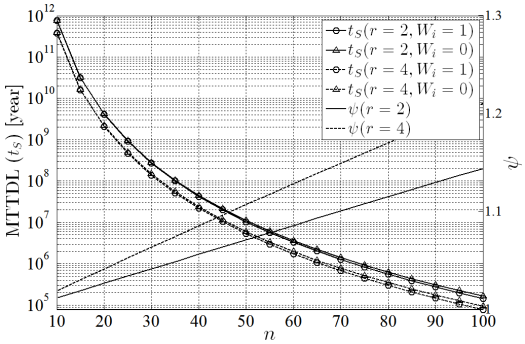

그림 2. 다양한 n, r, W_i에 대한 t_S와 ψ
Fig. 2. t_S and ψ versus n and various r, W_i

전체 노드 수 $M=400$, 저장 용량 $S=16\,TB$, 노드 1개의 복구 대역폭 $B=0.1\,Gbps$, 노드 1개 고장률 $\lambda = 4\,year^{-1}$, 시스템의 고장 검사 주기 $T_d = 30\,min$, 상태 i의 고장률 $\lambda_i = i\lambda$, 노드 복구율 $\rho_{1,i} + \rho_{2,i} = 1/T_d$, $\rho_{1,n-1} = (M-1)B/(Sr)$로 설정하였다. 이때 r은 평균 부분 접속 수다.

그림 2는 앞선 시스템 파라미터를 갖는 시스템의 n, r, W_i에 따른 t_S와 ψ 값을 나타낸다. 실험 결과 t_S는 [4]와 같이 n, r이 증가할수록 감소하는 경향을 보인다. ψ는 n과 r 각각에 대해 선형 관계에 근사함이 확인되었다. 이러한 현상은 노드 복구 중 이중 노드 복구 수행 횟수의 비율로 설명될 수 있다. n이 증가할수록 그림 1의 활동 상태 수가 증가하여 이중 노드 복구 수행 횟수 비율이 증가한다. 또한 r이 증가하면 이중 노드 복구의 유일한 단일 노드 복구율인 $\rho_{1,n-1}$의 값이 줄어 이중 노드 복구 수행 횟수 비율이 증가한다. 두 경우 모두 이중 노드 복구 수행 횟수를 증가시켜 ψ값이 증가하게 된다.

Ⅳ. 결 론

본 논문에서는 이중 노드 복구가 가능한 분산 저장 시스템의 MTTDL을 닫힌 형태의 식으로 보였다. 그리고 단일 노드 복구 대비 이중 노드 복구 시 MTTDL은 부호 길이와 평균 부분 접속 수에 선형 근사함을 확인하였다.

References

[1] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, “On the locality of codeword

symbols," *IEEE Trans. Inf. Theory,* vol. 58, no. 11, pp. 6925-6934, Nov. 2012.

[2] M.-Y. Nam and H.-Y. Song, "Constructions for optimal binary locally repairable codes," *J. KICS*, vol. 41, no. 10, pp. 1176-1178, Oct. 2016.

[3] S. Ramabhadran and J. Pasquale, "Analysis of long-running replicated systems," in *Proc. IEEE INFOCOM,* pp. 1-9, Barcelona, Spain, Apr. 2006.

[4] M. Shahabinejad, M. Khabbazian, and M. Ardakani, "A class of binary locally repairable codes," *IEEE Trans. Commun.,* vol. 64, no. 8, pp. 3182-3193, Aug. 2016.

[5] R. Bhagwan, K. Tati, Y.-C. Cheng, S. Savage, and G. M. Voelker, "Total recall: System support for automated availability management," in *Proc. NSDI*, pp. 337-350, CA, USA, Mar. 2004.