

데이터 접근 기록 정보를 이용한 적응적 데이터 복제 기법 제안

성기문*, 이동우, 최지현, R.S.Ramakrishna
광주과학기술원 정보통신공학과
e-mail:{gimun*, leepro, jhchoi80, rsr}@kjist.ac.kr

Adaptive Data Replication Strategy using data access history in DataGrid

GiMun Sung*, DongWoo Lee, Jihyun Choi, R.S.Ramakrishna
Department of Information and Communication,
Kwangju Institute of Science and Technology

요 약

프로세서 자원, 데이터 저장장치 자원을 제공하면서 가상기관(Virtual Organization)을 구성하는 각 사이트는 사용할 수 있는 네트워크 자원이 한정된 상황에서 애플리케이션 처리량을 극대화하는 최적화된 데이터그리드 시스템을 기대한다. 본 논문에서는 크기가 제한적이며 지리적으로 분산된 데이터 저장공간에서 적응적 데이터 복제 기법을 제안하고 Replica의 지리적 분배를 위한 평가 모델을 제안한다. 이를 위해 논리 시간 데이터 접근 기록 및 통계를 적용하여 복제할 파일들을 구분 하는 이산적 결정 모델을 제안하고 삭제할 Replica 결정에 논리 시간 접근 기록을 적용한다.

1. 서론

데이터그리드에서는 원격지 분산된 데이터의 접근에 드는 비용 최소화를 통해서 가상기관(VO[1])상의 애플리케이션 처리량 극대화를 기대할 수 있다.

이를 위해서 지리적으로 분산되어 있고 용량이 제한된 저장장치에 시간상으로 접근 빈도(temporal locality)가 높은 파일을 지리적인 면에서 접근이 용이한 곳에 replica를 생성 및 분배하는 것이 훌륭한 접근 방법이라 할수있다.

본 논문에서는 가상기관을 구성하는 각 사이트의 저장장치 공유 자원에 대해서 각 replica의 시간적인 접근 빈도 측정을 위해 보편적으로 사용되고 있는 데이터 접근 기록 기법을 사용한다. 이와 관련해 데이터 접근 빈도 측정을 위해 공유 저장장치를 제공하는 모든 사이트 별로 논리 시간(Logical Clock) 사용을 제안한다. 개별 사이트마다 데이터 접근 빈도 정보는 어느 파일을 복제할 것인지 결정하기 위해서 사용되고 통계적으로 언제 파일을 복제할 것인지 결정하기 위한 정보로도 활용된다.

한 사이트의 자율적 복제 결정방안에 따라 어떤 파일의 복제 생성이 결정되면 Replica를 어느 사이트에 복제하여야 시스템 처리량 극대화를 추구할 수 있을지는 절대적인 시스템 정보가 가용하지 않기 때문에 풀 수 없는 문제이다. 이에 가상기관 상에 사용할 수 있는 연결 네트워크 자원에 대한 정보와 해당 파일의 원격 사이트 접근 빈도정보를 바탕으로, 잠재적인 접근 사이트의 데이터 접근으로 기인하는 네트워크 링크점유 시간을 최소화시키는 Replica 목적 사이트를 찾아 Replica를 복제하는 모델을 제안한다.

2. 관련 연구

DCDS[2]에서는 계층적 네트워크 구조에서 계층적으로 분산된 데이터 접근비용을 최소화하는 Job Scheduling과 데이터 접근 테이블을 바탕으로 한 독립적인 데이터 복제가 시스템 처리량 극대화를 필연적으로 가져온다고 소개했다. 그러나 계층적인 네트워크 구조는 한정된 가상기관을 위해서만 유효하고 원격 데이터가 최상위 링크에 존재한다는 조건하에 성

능평가가 이루어졌다.

Economy-Based File Replication Strategy[3]에서는 P2P기반의 경매프로토콜을 바탕으로 각 사이트별로 데이터 접근 요청 메시지 정보 리스트를 이용해서 경제적으로 이득이 된다고 판단될 때 수신자가 복제를 발의하는 구조를 따르고 있다. 그러나 접근 요청 메시지 정보 리스트를 바탕으로 한 가치 평가 전제로 특정한 사용자 접근 패턴을 가정하고 있기 때문에 보편적인 애플리케이션 적용시 효율성 문제가 야기된다.

본 논문에서는 DCDS논문의 데이터 접근 테이블을 바탕으로 한 Replica 제공자 발의의 복제 구조를 따른다. 이에 논리 시간 데이터 접근 기록 집합 관리기법과 Replica결정을 위한 임계치 및 논리 시간 창 관리방안 그리고 적응적 이산적 복제 최적화 모델을 제안한다.

3. 논리 시간 데이터 접근 기록

논리 시간을 사용을 통한 이점으로는 데이터 접근 기록 순서 작성에 있어 실제 시각을 사용하는 것보다 적은 비용으로 테이블을 구성할 수 있고 기록 작성 및 이를 해석함에 있어 보다 융통성 있는 정보를 얻어낼 수 있다.

Replica 생성 의사 결정을 위해서 사이트 내부와 외부 접근 정보를 구분할 필요가 있다. 외부 접근 빈도는 Replica 생성을 위한 정보로 사용한다. 내부 접근 빈도는 Replica 생성을 위한 정보로는 사용하지 않지만 Replica 삭제를 위한 정보로 사용한다. 내부와 외부 접근을 구분하기 위해 홀수 논리 시간은 내부 접근 기록에 사용하고 짝수 논리시간은 외부 접근 기록에 사용한다.(표1. 참조) 단, 홀수 논리 시간은 재사용할 수 있게한다. 즉 증가시키지 않고 재사용한다.

- Replica 생성 정보 : 짝수 논리 시간만 사용.
- Replica 삭제 정보 : 모든 논리 시간 사용.

$$\bullet \frac{\sum \text{Odd Logical Clock}}{\sum \text{Even Logical Clock}} = \frac{\text{Local}}{\text{Remote}} \text{Access Ratio}$$

4. Replica 삭제 방안

Replica 생성 의사결정은 각 사이트의 이산적 복제 결정 프로세스에 의해 이루어진다. 실제 생성될 Replica를 수용할 사이트에서 저장장치 용량이 부족한 경우 기존의 파일 중에 미래 사용 효율성이 떨어지는 파일을 선택하여 제거해야한다. 어떤 파일의 미래 사용 가치의 판단 척도로 앞으로 지역 사이트내의 데이

Replica ID	Logical Clock Access History				# of History	Remote Access #
	Access Site ID				A Replica rate for deletion since Rj_old	
R12	125	150	161	170	4	2
	Sj	S5	Sj	S2	4/(200-125)≈1/19	
R30	50	100	134		3	3
	S5	S3	S10		3/(200-50)=1/50	
R23	198				1	1
	S3				1/(200-198)=1/2	
R54	187				1	0
	Sj				1/(200-187)=1/13	
...	...					
	Rj_old				200 == Present Logical Clock	

표 1. Sj 사이트의 논리 시간(Logical Clock) 테이블 예
터 접근 기대치(e_L)와 원격지 사이트의 데이터 접근 기대치(e_R)로 구분된다.(차별 해석 가능 : e_L↔e_R)

데이터 복제 방법론의 성능 평가를 위해 사용할 수 있는 통합된 Replica 삭제 파일 결정 방안으로 논리 시간 접근 기록을 이용하면 아래와 같이 분류 및 적용할 수 있다.

4.1 Least recently used file replacement

- Replica 삭제 파일 결정을 위해서 가장 오래전에 사용한 파일을 선택한다. 즉 논리 시간 테이블에 각 Replica에 대해 접근 기록상의 제일 끝 논리 시간이 가장 작은 Replica를 지운다.

4.2 Least time accessed file replacement

- 각 Replica에 대한 논리 시간 접근 기록 수가 가장 작은 값을 선택한다. 같은 기록 수를 가지는 Replica가 여러 개 있으면 사용한 지 가장 오래된 파일을 선택(4.1)한다.

- 최근에 생성된 Replica를 지우지 않기 위해서 Replica 생성시 지역 내의 데이터 접근시 사용하는 홀수 논리 시간(현재 논리 시간이 홀수이면 이를 재사용하고 짝수이면 하나 증가해서 사용한다)을 평균 접근 수(M)만큼 초기화한다.

4.3 Least rate accessed file replacement since each oldest logical time

- 각 Replica(R_j)의 가장 오래된 논리 시간 이후부터 현재의 논리 시간까지 해당 파일에 대한 접근 빈번함을 Replica 삭제 파일 결정 평가 잣대로 사용한다.

$$\frac{\text{Number of Logical Clock History}_{R_j}}{\text{Present Logical Clock} - \text{Oldest Logical Clock}_{R_j}}$$

- 표 1(_A Replica rate for deletion since Rj_old) 참조

4.4 Least diverse weighted-accessed file replacement

- 데이터 접근 기록 시간이 현재 논리 시간으로부터 얼마나 떨어져 있는가에 따라 각 기록 사건에 대해서 영

향력을 변화해서(아래 weight function을 이용) 각 Replica(R_j) 가치 평가를 위한 정보로 사용한다.

- Weight Function Samples

$$(a) \sum_{i=1}^{\#of LC_{R_j}} \frac{1}{Present Logical Clock - Logical Clock_i}$$

$$(b) \sum_{i=1}^{\#of LC_{R_j}} \frac{Logical Clock_i - L.C.Oldest_{Site}}{Present Logical Clock - L.C.Oldest_{Site}}$$

$$(c) \sum_{i=1}^{\#of LC_{R_j}} (0.1 * EXP\{x\} - 0.1) \quad \text{where}$$

$$x = \frac{2.4 * (Logical Clock_i - L.C.Oldest_{Site})}{Present Logical Clock - L.C.Oldest_{Site}}$$

- L.C.Oldest_{Site}는 각 사이트 기록 테이블의 최소 논리 시간임
 - PresentLogicalClock - L.C.Oldest_{Site} ≡ 논리 시간 창 크기

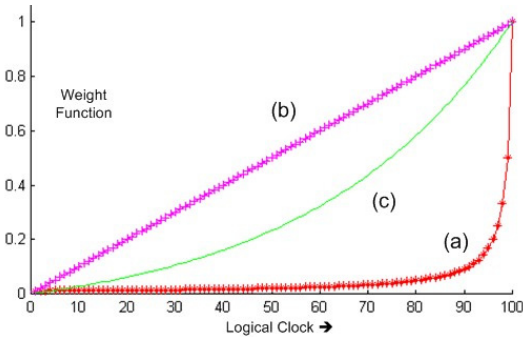


그림 1 Weight Function (a),(b),(c)

5. 복제 임계치 와 논리 시간 창 최적화 프로세스

IDRS[4]에서도 언급되었듯이, 데이터 접근 기록 및 운용을 통한 Replica 관리는 복제 임계치 최적화 프로세서가 시스템 처리량에 직접적으로 영향을 준다. 가상기관 내의 각 사이트는 사용자들의 변화하는 접근 패턴을 수용하며 시스템 처리량 극대화를 도모하기 위해서 복제 임계치의 적응 값을 찾아가야 한다.

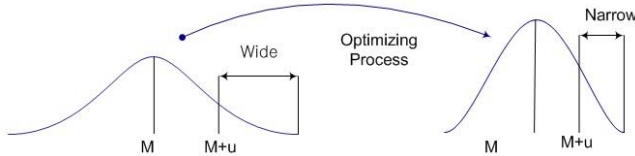


그림 2. Replica 생성파일 결정 프로세스의 최적화 과정 모델

그림 2.는 복제 임계치 최적화 프로세스 과정을 보여준다. 각 사이트의 Source 혹은 Replica들의 접근 빈도에 따라 통계적으로 평균값(M)과 분산(σ²)정보가 변화한다. 따라서 복제 임계치 최적화 프로세스 모델의 임계치 값으로 (M, u=g(σ,M))를 제안한다. 각 사이트별 표준편차(σ)는 Replica 파일들의 접근 빈도와 패턴에 맞추어 적응적으로 복제가 진행됨에 따라 작

아지게 된다. 따라서 복제 최적화 과정은 그림 2. 좌의 상태에서 우의 상태로 변화한다. dt_i = f(M_i, σ_i)의 이산적인 시간간격마다 (M,u)_i를 제 평가하고 Replica 임계치를 초과하는 Source/Replica에 대해서 새로운 Replica를 생성하는 최적화 프로세스 모델을 제안한다.(그림 3. 참조)

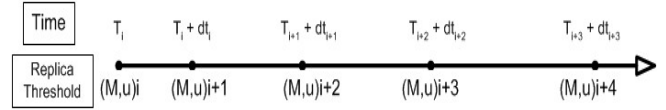


그림 3. 이산적 복제 결정 시간 간격 프로세스

논리 시간 창 크기(CWS)는 데이터 접근 변화에 대한 민감도를 높이기 위해서는 크기를 줄여야 하고 민감도를 낮추기 위해서는 크기를 늘려야 한다. 다시말해 데이터 접근 변화에 응답특성은 떨어지지만 Replica 생성의 정확도를 높이려면 논리 시간 창 크기를 늘려야 하고, 반면에 빠른 응답특성을 원한다면 논리 시간 창 크기를 줄여야 한다.

$$- StorageRatio_j(SR_j) = \frac{Storage Space_j}{Standard Storage Size}$$

$$- EffectiveIORatio_j(EIOR_j) = \frac{Effective IONet_j}{Standard IONet}$$

$$- StaticClock WindowSize_j(SCWS_j) = Standard CWS * SR_j * EIOR_j$$

$$- DynamicCWS_j(DCWS_j) = SCWS_j * DynamicScaleFactor_j$$

※ Standard CWS는 가상기관의 애플리케이션 접근 특성을 반영할 수 있도록 설정 되어야 한다.

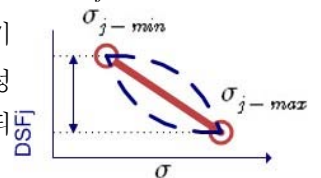


그림 4. DynamicSFj 조정 모델

이상의 복제 최적화 프로세스의 시스템 상호작용 모델은 다음과 같다.

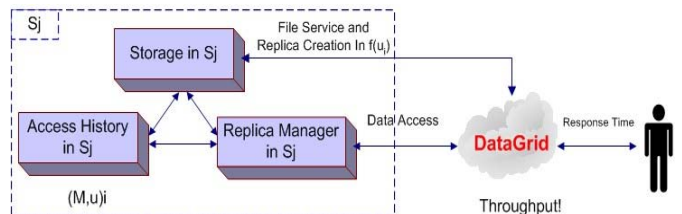


그림 5. 복제 최적화 프로세스 시스템 상호작용 모델

6. 데이터 접근 점 지향의 Replica 생성 위치 결정 방안

어떤 파일의 Replica 생성이 결정되면 해당 파일 (R_j)의 접근 기록 테이블을 이용해서 Replica를 수용

할 사이트를 결정해야 한다. 미래에 잠재적인 데이터 접근 요구에 대해서 최소의 비용(정의→네트워크 링크 자원 점유 시간)으로 서비스할 수 있는 지점을 찾는 것이 Replica 생성 위치 결정 방안이다.

이를 위해서 철저히 해당 파일의 접근 기록과 시스템 차원에서 가용한 매개변수를 이용해서 결정해야 한다. 그 매개변수로 가상기관 애플리케이션에 가용한 네트워크 대역폭 정보와 사이트 간의 연결 정보를 바탕으로 최소의 네트워크 링크 점유 시간(Link Use Time)을 평가 잣대로 사용한다.

- 단위 데이터를 전송함에 있어 각 네트워크 링크 단위

$$\text{점유되는 시간} : L.U.T_i = \frac{1}{Band\ Width_i} \left[\frac{Secs}{MBytes} \right]$$

- Replica 생성 위치 후보 사이트(Sc)에서 데이터 접근 요구 사이트까지 링크를 점유하는 시간 : Path_j

$$Path_j = \sum_{i=1}^{\#of\ hops\ to\ Access\ Site_j} \frac{1}{Band\ Width_i}$$

- 사이트 S_s로 부터 { Site_j | Site_j ∈ {R_k Access History Set} = R_kAHS } 에 대해 w_j(w_j는 R_k Replica 에 Site_j의 접근 기록 수)번씩 서비스 할 때 Link Use Time :

$$ExLUT(S_s, R_k) = \sum_{j=1}^{\#of\ Site \in R_k AHS} w_j * Path_j$$

$$Smallest\ ExLUT(S_s, R_k) = ExLeastLUT(S_s, R_k) = \sum_{j=1}^{\#of\ Site \in R_k AHS} w_j * Path_{j-Min}$$

- Site_j ∈ R_kAHS에 대해서 w_j번씩 Path_{j-Min(S_s, Sc)} (Path_{j-Min(S_s, Sc) = Min(Path_{j-Min-S_s}, Path_{j-Min-Sc}))을 따라서 서비스 할 때 Internetwork Link Use Time :}

$$CoLeastLUT(S_s, Sc, R_k) = \sum_{j=1}^{\#of\ Site \in R_k AHS} w_j * Path_{j-Min\{S_s, Sc\}}$$

G=(V,E) 그래프에서 두 노드 간에 Complexity가 O(N²)인 shortest path algorithm을 이용하면 CoLeastLUT(S_s, Sc, R_k)을 구하기 위한 복잡도는 O(N³)이 된다.

- 표 2. 사용된 Source/Replica R_k의 접근 기록 정보

S _j	S1	S2	S7	S11	S16	S18	S20
Weight	1	2	1	2	1	1	2

- CoLeastLUT(S_{S_s=S10}, Sc, R_k)의 순위 결과

- < S10 , S21 > : Weight_Sum=2.155
- < S10 , S20 > : Weight_Sum=2.166
- < S10 , S19 > : Weight_Sum=2.182
- < S10 , S18 > : Weight_Sum=2.272

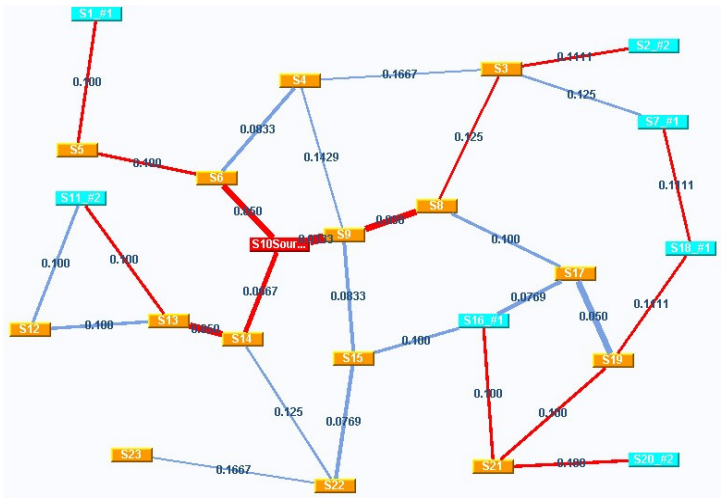


그림 6 CoLeastLUT(S10, Sc, R_k) 시뮬레이션 결과
● < S10 , S16 > : Weight_Sum=2.293

7. 결론 및 향후과제

제한된 저장장치 자원과 네트워크 자원을 바탕으로 애플리케이션 처리량 지향의 최적화된 데이터그리드를 위해 효율적인 복제 방법론이 절실히 필요하다.

본 논문에서 데이터 접근 기록 정보를 효율적으로 기록 해석하여 시간적인 면과 지리적인 면을 동시에 다루는 적응적 데이터 복제 해법을 제시했다.

향후과제로 CoLeastLUT 평가와 네트워크 연결구조를 고려한 TCoLeastLUT에 대한 연구 그리고 최적화 파라미터 (M, u=g(σ,M)), dt_i = f(M_i, σ_i), DynamicCWS_j 관계모델 수식화 및 평가를 위해 시뮬레이션 구현예정에 있다.

참고문헌

[1] Forster, I., Kesselman, C. and Tuecke, S. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations." International Journal of High Performance Computing Applications, 15(3). 200-222
 [2] Kavitha Ranganathan, Ian Foster, "Decoupling Computation and Data Scheduling in Distributed Data-Intensive Applications." 11th IEEE International Symposium on High Performance Distributed Computing.
 [3] William H. Bell 1 , David G. Cameron 1 , Ruben Carvajal-Schiaffino 2 , A. Paul Millar 1 "Evaluation of an Economy-Based File Replication Strategy for a Data Grid." in International Workshop on Agent-based Cluster and Grid Computing at CCGrid 2003, May 2003. Scheduling Optimisation Strategies
 [4] Kavitha Ranganathan and Ian Foster "Identifying Dynamic Replication Strategies for a High-Performance Data Grid", Proceedings of the International Workshop on Grid Computing, Nov 2001.