

고효율 중복제거 기반 백업 시스템을 위한 실제 서버 환경에서의 파일 스터디

윤대영^o 원유집

한양대학교 전자컴퓨터통신공학과 임베디드 소프트웨어 시스템 연구실
dayyoung@ece.hanyang.ac.kr, yjwon@ece.hanyang.ac.kr

File study for Highly Efficient Deduplication Backup system

Daeyoung Yoon^o Youjip Won

ESOS Lab. Dept. of Electronics and Computer Engineering, Hanyang University

요 약

급증하는 데이터의 증가 추세에 맞춰서 백업의 중요성이 점차 대두되고 있다. 본 논문에서는 중복제거 기반 백업 시스템 환경에서 파일들의 특성을 파악하여 최적화된 성능을 보일 수 있는 방향을 제시한다. 본 논문에서는 중복감지 범위에 따라 세션 내 중복감지 알고리즘, 증분식 중복감지 알고리즘 그리고 전후-세션 중복감지 알고리즘으로 구분하였고 이들 3 가지 알고리즘을 이용하여 실제 중복감지율의 변화를 측정하였고 그 결과를 토대로 중복감지 효율성에 대하여 분석한다.

1. 서 론

오늘날 디지털 데이터의 양은 기하급수적으로 증가하고 있으며 이러한 디지털 데이터를 저장 및 관리하는 스토리지 시스템의 중요성은 점점 커져가고 있다[1]. 또한 정보의 비대화와 함께 정보의 손실에 대한 대비도 중요하게 되어 백업의 중요성은 날이 대두되고 있다.

중복제거 기반 백업 시스템은 중복된 데이터는 저장하지 않고 새로운 데이터만 백업 저장매체에 저장하는 시스템이다[2][3][5]. 데이터의 중복성을 판별하기 위해서 청크라 불리는 파일 조각으로 파일을 분할하여 해당 청크의 해쉬값을 비교한다. 파일분할 알고리즘에는 크게 가변길이 파일분할 알고리즘과 고정길이 파일분할 알고리즘으로 나눌 수 있다[4][5]. 가변길이 파일분할 알고리즘은 중복성 검사에 용이하지만 파일 분할 시에 계산량이 많아 CPU에 부하가 많이 걸리고 전체 백업 속도에 영향 미친다. 반면 고정길이 파일분할 알고리즘은 별도의 계산량이 많지 않아서 파일분할 속도가 빠르지만 중복성 검사에 효율적이지 않다[5].

본 논문은 실제 서버 환경에서 파일들의 변화 추이를 추적한다. 중복감지 효율성을 실험하기 위해 중복감지 방법을 3 가지로 나누어 측정하였다. 세션 내 중복감지 알고리즘, 증분식 중복감지 알고리즘, 그리고 전후-세션 중복감지 알고리즘이다(각각의 알고리즘에 대한 설명은 단원 5에서 소개한다). 이들 3 가지 중복감지 방법을 통해 각각의 중복감지 효율성을 비교하였고 이를 통해 성능 개선 향상에 지표가 되는 정보를 제공한다.

2. 실험 구성 및 용어 정리

본 파일 스터디에 사용된 실험 환경은 Intel E540S 쿼드 코어, 메모리 8 GB, 13 TB(RAID5) + 13 TB(RAID0)로 구성된 백업 서버에 운영체제는 Red Hat 5.3 Enterprise version 4.1.2-44이다. 실험 데이터는 실제 서버환경에서 사용되고 있는 7.33 TB로 총 17 개의 서로 다른 서버에서 각각의 서버에 5일치 분량의 중복제거 되지 않은 풀 백업(full backup) 받은 자료이다.

세션(session): 특정 백업 시점에서 백업 대상이 되는 파일 집합. 세션은 ‘[세션][백업 날짜 순서]’ 형식으로 쓰인다. 예를 들어 ‘세션2’라 하면 두 번째 중복 검사를 통한 백업 시점에서의 백업 대상이 되는 파일 집합을 일컫는다.

패치(patch): 중복되지 않은 새로운 청크들의 집합

패치 비율(patch ratio): 패치에 속하는 청크 크기의 총합 ÷ 해당 세션에 속하는 파일들의 크기 총합

서버 이름	전체용량(GB)	세션 수
홈페이지 관리 서버	1375.9	5
e-learning database 서버	480.9	5
사무지원 database 서버	531.6	5
신입학 관리 database 서버	201.5	5
행정지원 서버1	530.6	5
행정지원 서버2	358.7	5
행정지원 서버3	705.7	5
행정지원 서버4	40.8	5

행정지원 서버5	26.1	5
행정지원 서버6	12.1	4
행정지원 서버7	177.0	5
행정지원 서버8	191.5	5
행정지원 서버9	129.5	5
메일 서버	325.3	5
메일 데이터 서버1	786.2	5
메일 데이터 서버2	790.5	5
메일 데이터 서버3	844.2	5
전체 실험 데이터	7509.0	84

<표1> 실험 데이터 정보

표1은 전체 실험 데이터의 용량 및 세션 수를 각 서버 단위로 조사한 결과이다. 전체 실험 데이터의 용량은 약 7509.02 GB이고 총 세션은 84개이다.

3. 중복 감지율 분석

각 날짜 별 세션과 세션 사이에 동일한 서버간의 실제 중복감지율을 분석하였다. 본 실험에서는 세션 내 중복감지 알고리즘(Intra-session Duplication Detection)과 증분식 중복감지 알고리즘(Incremental Duplication Detection) 그리고 전후-세션 중복감지 알고리즘(Lookahead Duplication Detection), 세 가지 알고리즘의 중복감지 효율성을 비교 분석하였다. 실험은 가변길이 파일분할 알고리즘을 사용하였고 평균 청크 크기는 9.96 KB로 나왔다.

3.1 세션 내 중복감지 알고리즘 적용 시 중복감지율

세션 내 중복감지 알고리즘(Intra-session Duplication Detection)은 백업 대상이 되는 파일 집합을 어느 한 시점에서 중복 제거를 통한 백업을 하고자 할 때 그 시점에 속하는 파일 집합 내에서만 중복된 청크가 존재하는지 비교, 검사하는 알고리즘이다.

중복감지율(단일-세션 중복감지 알고리즘)	
세션1	27.1 %
세션2	25.6 %
세션3	25.4 %
세션4	25.4 %
세션5	25.3 %

<표2> 세션 내 중복감지 알고리즘을 이용한 중복감지율

표2는 각 세션 별로 날짜 순서에 의하여 중복되는 청크들의 크기 총합을 계산하여 전체 세션의 용량으로 나눈 백분율이다. 세션1 내에서 중복된 청크로 판명된 청크들의 용량은 세션 전체 용량의 27.15 %이다. 세션2와 세션3, 세션4 그리고 세션5의 중복감지율은

25.65 %, 25.43 %, 25.47 % 그리고 25.37 %이다. 단일 세션 안에서 서로 다른 경로 또는 서로 다른 파일 이름을 가진 파일들 간의 중복감지율이 평균 25.82 %이다.

각 서버 별 중복감지율(세션 내 중복감지 알고리즘)	
홈페이지 관리 서버	26.07 %
이러닝 데이터베이스 서버	5.78 %
사무지원 데이터베이스 서버	4.28 %
신입학 관리 데이터베이스 서버	15.28 %
행정지원 서버1	31.75 %
행정지원 서버2	31.75 %
행정지원 서버3	22.54 %
행정지원 서버4	23.61 %
행정지원 서버5	32.70 %
행정지원 서버6	31.40 %
행정지원 서버7	34.58 %
행정지원 서버8	37.29 %
행정지원 서버9	31.80 %
메일 서버	8.88 %
메일 데이터 서버1	34.10 %
메일 데이터 서버2	33.31 %
메일 데이터 서버3	33.10 %

<표3> 세션 내 중복감지 알고리즘을 이용하여 각 서버 별 중복감지율 조사

표3은 각 서버의 세션 내 중복감지 알고리즘을 사용하여 파일분할을 실행했을 때 감지되는 중복된 청크들의 용량 비율을 조사한 결과이다. 3개의 데이터베이스 서버와 메일 서버는 여타 다른 서버들에 비해서 낮은 중복감지율을 나타낸다. 신입학 관리 데이터베이스 서버를 제외한 나머지 2개의 데이터베이스 서버는 각각 5.78 %와 4.28 %의 가장 낮은 중복감지율을 나타내고 있다. 신입학 관리 데이터베이스 서버의 경우 15.28 %의 중복감지율을 보이고 있는데 이는 해당 서버의 첫 번째 세션(세션1)에서 기존 내용과 완전히 동일한 복사본 파일이 하나씩 더 존재하기 때문이다. 신입학 관리 데이터베이스 서버의 세션1에서의 중복감지율은 44.15 %이고 나머지 세션2, 3, 4, 5는 모두 0.69 %로 동일하다. 메일 서버의 중복감지율은 8.88 %이다. 메일 서버에는 메일 서비스를 실행하기 위한 각종 유틸리티와 운영체제가 저장되어 있고 실제 메일 파일은 메일 데이터 서버에 저장되어 있다. 메일 서버의 중복감지율이 낮은 이유는 실행 및 압축 형태의 파일들이 대부분인 시스템 파일로 주로 구성되어서 중복률이 낮다. 세션 내 중복감지 알고리즘 적용 시에 각 서버의 특성에 따라서 중복감지율의 차이가 난다. 따라서 세션 내에서의 청크의 중복감지율은 특수한

경우를 제외하고 (신입학 관리 데이터베이스 서버) 서버의 특성에 따라 의존적임을 알 수 있다.

3.2 증분식 중복감지 알고리즘 적용 시 중복감지율

증분식 중복감지 알고리즘(Incremental Duplication Detection)은 이전에 백업 되었던 모든 세션의 청크 정보를 유지하여 현재 백업 대상이 되는 세션과 이미 백업 서버에 저장된 모든 세션의 청크 정보를 비교하여 중복된 청크인지 여부를 판별하는 알고리즘이다. 일반적으로 대다수의 중복제거 기반 백업 시스템에서는 증분식 중복감지 알고리즘을 채용하고 있다.

패치 비율(증분식 중복감지 알고리즘)	
세션1	72.8 %
세션2	1.1 %
세션3	1.0 %
세션4	1.1 %
세션5	1.1 %

<표4> 증분식 중복감지 알고리즘을 이용한 패치 비율

표5는 증분식 중복감지 알고리즘을 이용하여 각 세션의 파일들을 중복제거 기반 백업 시스템을 통하여 백업할 때마다 저장매체에 새로이 저장되는 청크들의 용량을 해당 세션의 총용량으로 나눈 비율을 조사한 결과이다. 3.1 단원과 달리 중복감지율 대신 패치 비율(patch ratio)로 나타난 이유는 세션과 세션간에 중복 비교하여 제거 시에 실제 저장매체 소모율을 분석하기 편하게 하기 위해서이다. 세션1을 제외한 나머지 4 개 세션의 패치 비율은 1 % 조금 넘는다. 세션1에서 72.84 %가 나온 이유는 해당 세션에 속하는 파일들 간의 중복률이 27.16 %가 나와서 새로이 저장되는 청크들의 용량이 72.84 %가 되었다. 세션2 이후부터는 그 전 세션들의 청크 정보와 비교하여 중복 여부를 판단하기 때문에 중복률이 99 %에 가깝게 나온다.

각 서버 별 중복감지율(증분식 중복감지 알고리즘)	
홈페이지 관리 서버	84.62 %
이러닝 데이터베이스 서버	79.22 %
사무지원 데이터베이스 서버	78.81 %
신입학 관리 데이터베이스 서버	80.84 %
행정지원 서버1	86.30 %
행정지원 서버2	86.82 %
행정지원 서버3	83.86 %
행정지원 서버4	84.72 %
행정지원 서버5	86.54 %
행정지원 서버6	82.85 %
행정지원 서버7	86.91 %

행정지원 서버8	87.45 %
행정지원 서버9	86.36 %
메일 서버	78.81 %
메일 데이터 서버1	86.18 %
메일 데이터 서버2	85.98 %
메일 데이터 서버3	85.92 %

<표5> 증분식 중복감지 알고리즘을 이용한 중복 감지율

표5는 증분식 중복감지 알고리즘을 이용했을 때 각 서버 별로 중복감지율을 측정된 결과이다. 세션1을 제외한 세션2, 3, 4, 5의 평균 중복감지율은 98.86 %로 매우 높은 중복률을 나타낸다. 이 실험에서 서버의 특성에 따라서 중복감지율 변화가 비교적 있는 세션 내 중복감지 알고리즘과 달리 증분식 중복감지 알고리즘은 서버의 특성에 영향 받지 않는다. 증분식 중복감지 알고리즘은 첫 세션을 제외한 나머지 세션에서 높은 중복감지율을 보인다.

3.3 전후-세션 중복감지 알고리즘 적용 시 중복감지율

전후-세션 중복감지 알고리즘(Lookahead Duplication Detection)은 현재 백업 대상 시점과 시간적으로 가까운 시점의 세션에 속한 청크 정보와 현재 백업 대상이 되는 세션의 청크들을 비교하여 중복여부를 판단하는 알고리즘이다. 이 알고리즘은 해당 세션 내에서 서로 다른 파일간의 중복은 중복 감지 대상에 포함되지 않는다.

패치 비율(전후-세션 중복감지 알고리즘)	
세션1	100 %
세션2	1.2 %
세션3	1.1 %
세션4	1.2 %
세션5	1.2 %

<표6> 전후-세션 중복감지 알고리즘을 이용한 패치 비율

표6은 전후-세션 중복감지 알고리즘을 이용하여 패치 비율을 계산한 결과이다. 세션1을 제외한 세션2, 3, 4, 5의 패치 비율은 앞서 3.2 단원에서 논의 되었던 증분식 중복감지 알고리즘을 적용 했을 때와 차이가 거의 없다. 세션1에서 중복된 청크가 전혀 발견되지 않은 이유는 비교할 청크 정보가 없어서 세션1에서 생성되는 모든 청크는 중복 비교 검사 없이 새로운 청크로 판명되기 때문이다.

청크의 중복 여부를 파악하기 위해서는 청크의 해쉬값인 요약정보(fingerprint)를 백업 서버에 기 저장된 청크 요약정보와 비교하여 해당 청크를 백업 대상에서 제외할 지 여부를 결정한다. 증분식 중복감지 알고리즘이 적용된 백업 시스템 환경에서는 이전에

백업된 모든 청크 요약 정보 히스토리를 관리하기 때문에 백업 된 세션 수가 늘어남에 따라 관리해야 하는 청크 요약정보의 수가 기하급수적으로 늘어난다. 이는 청크의 중복 검사를 위해 소요되는 시간이 급격히 늘어나게 되어서 전체 성능 저하에 결정적인 원인이 될 수 있다. 따라서 청크의 중복 검사 속도가 급격하게 저하되는 것을 방지하기 위해서는 백업 서버에서 관리해야 하는 청크 요약정보의 수를 효과적으로 조절해야 한다. 따라서 백업된 세션 수가 매우 많고 대용량일 때는 전후-세션 중복감지 알고리즘이 증분식 중복감지 알고리즘보다 더 효율성을 나타낸다. 반면에 전후-세션 중복감지 알고리즘은 첫 번째 세션(세션1)에서 중복 검사를 전혀 하지 않아서 중복감지율이 0 %인 초반 저장매체 소모 손실이 상대적으로 크다. 또한 매 세션의 중복감지율이 극소하지만 증분식 알고리즘에 비해 낮아서 백업 주기가 매우 길거나 세션의 규모가 작을 경우 성능 기대 효과가 그리 크지 않고 오히려 중복감지율만 떨어져서 저장매체 소모율이 크게 증가하는 결과가 나올 것이다.

각 서버 별 중복감지율(전후-세션 중복감지 알고리즘)	
홈페이지 관리 서버	79.39 %
이러닝 데이터베이스 서버	77.78 %
사무지원 데이터베이스 서버	77.90 %
신입학 관리 데이터베이스 서버	65.97 %
행정지원 서버1	79.95 %
행정지원 서버2	78.53 %
행정지원 서버3	79.35 %
행정지원 서버4	80 %
행정지원 서버5	80 %
행정지원 서버6	75 %
행정지원 서버7	80 %
행정지원 서버8	79.99 %
행정지원 서버9	80 %
메일 서버	77.09 %
메일 데이터 서버1	79.15 %
메일 데이터 서버2	79.11 %
메일 데이터 서버3	79.09 %

<표7> 전후-세션 중복감지 알고리즘을 이용한 중복감지율

표7은 전후-세션 중복감지 알고리즘을 이용하여 각 서버의 세션간의 중복률을 조사하였다. 평균 중복감지율은 78.67 %이고 항상 0 %의 중복감지율을 보이는 세션1을 제외한 4개 세션만의 평균 중복감지율은 98.77 %로 증분식 중복감지 알고리즘의 중복감지율과 거의 차이가 없다.

6. 결 론

본 논문에서는 7.33 TB 규모의 실제 서버에 저장된 파일들을 이용하여 중복제거 기반 백업 시스템의 성능을 개선시키기 중복감지율과 그 속도에 대해서 분석하였다.

세션 내 중복감지 알고리즘은 단일 세션의 청크 정보만을 유지하고 중복 검사하고자 하는 청크와 기 저장된 청크의 유일성을 비교한다. 그 결과 평균 중복감지율은 25.82 %가 나왔다. 증분식 중복감지율 알고리즘은 중복 검사를 마친 모든 세션의 청크 요약정보 히스토리를 백업 서버에 저장하여 중복 비교 시에 활용하는 알고리즘이다. 이 알고리즘 적용 시 평균 중복감지율은 84.28 %로 가장 높은 중복감지율을 보인다. 전후-세션 중복감지 알고리즘은 가장 최근에 중복 검사를 한 세션의 청크 요약정보만 백업 서버에서 유지한다. 따라서 중복 검사는 바로 이전에 백업된 세션과 비교하여 중복되었는지 유무만 파악한다. 평균 중복감지율은 98.77 %이다.

Acknowledgement

This work is sponsored by the IT R&D Program(large scale hyber-MLC SSD development, No. 10035202 of MKE/KEIT, Korea, National Research Lab grant.

참고문헌

- [1] J. McKnight, T. Asaro, and B. Babineau, "Digital Archiving:End-User Servey and Market Forecast 2006-2010," The Enterprise Strategy Group Jan. 2006.
- [2] Y. Won, J. Ban, J. Min, J. Hur, S. Oh, and J. Lee, "Efficient index lookup for de-duplication backup system," in Modeling, Analysis and Simulation of Computers and Telecommunication Systems, 2008. MASCOTS 2008. IEEE International Symposium on (Poster Presentation), Sept. 2008, pp. 1-3
- [3] Y. Won, R. Kim, J. Ban, J. Hur, S. Oh and J. Lee, "Prun : Eliminating information redundancy for large scale data backup system," IEEE International Conference on Computational Sciences and Its Application(ICCSA `08), Perugia, Italy, 2008.
- [4] D. Meister and A. Brinkmann, "Multi-level comparison of data deduplication in a backup scenario," in Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference. Haifa, Israel: ACM, May 2009, pp. 1-12.
- [5] J. Min, D. Yoon, Y. Won, "Efficient Deduplication Techniques for Modern Backup Operation," IEEE Transaction on Computer(TC), Oct. 2010.