

이종 시스템 통합을 위한 데이터 관리 기법 모델링

강인성*, 이홍철**

***고려대학교 정보경영공학전문대학원
e-mail:isk917@korea.ac.kr

Data management Scheme modeling for Heterogeneous system integration

In-Seong Kang*, Hong-Chul Lee**

***Graduate School of Information Management & Security,
Korea University.

요 약

본 논문에서는 Ubiquitous Computing 환경 하에서 이종 시스템 간의 통합을 위한 데이터 관리 기법 모델을 제안하였다. 이종 시스템 간의 통합이 이루어지면 방대한 양의 데이터를 모든 시스템이 공유해야 하기 때문에 무분별한 데이터의 중복과 저장으로 인해 프로세스의 데이터 처리 성능 및 데이터 무결성을 보장받지 못 하는 등의 문제점이 발생한다. 이를 보완하기 위해 Minimal cost Spanning tree의 원리를 적용하여 시스템 통합에 따른 데이터 처리 및 무결성 문제 해결을 위한 메커니즘을 제시하고자 한다.

1. 서론

Ubiquitous Computing 환경에서는 빠르게 변화되는 수많은 데이터를 어떠한 제약에도 구애받지 않고 수집가능하나 수집이상으로 중요한 것이 바로 관리다. 데이터가 의사결정에 유효한 영향을 미칠 때 더욱 가치를 인정받을 수 있기 때문에 보다 나은 의사결정을 위해 효과적인 데이터 관리에 대한 다양한 연구가 이루어지고 있다.

대부분의 시스템 의사결정은 시스템 A가 수행하는 의사결정이 시스템 B나 C의 상황 변화에 영향을 주고, 시스템 B나 C의 상황 정보가 시스템 A에게 실시간으로 전달될 수 있음을 기본으로 한다. 이러한 이종 시스템 간의 문제 해결 방법 중의 하나는 모든 데이터를 공유하는 것이다.

그러나 데이터 공유 방식은 다음과 같은 문제점들이 내포되어 있다. 첫째, 모든 시스템이 공통적으로 방대한 데이터를 모두 보유해야 하기 때문에 처리 속도에 영향을 줄 수 있다. 둘째, 개별 시스템이 처한 상황 변화로 인해 공유 데이터를 항상 변경해야 할 경우 일관성 있는 변경이 어렵다. 그리고 마지막으로 공유 데이터의 무결성이 보장되지 않아 문제 해결에 어려움이 발생할 수 있다.

갈수록 컴퓨터를 이용한 프로세스가 증가하고, 시스템 간의 데이터를 공유하는 것이 일반화되면서 시스템 통합(system integration)이 중요시되고 있다. 또한 이로 인해 발생하는 데이터 처리에 대한 분야 역시 중요하게 다뤄지고 있다. 특히 데이터 중복현상은 동등 계층 통신(P2P: peer to peer) 시스템, 백업(backup) 시스템, FTP미러(mirror) 그리고 가상화(virtualization) 시스템 등에서 높은 비율로 발생되고 있다. 예를 들면 리눅스 FTP미러의 저장 서버에서 동일한 파일을 다양한 종류의 미디어 포맷으로 변환(CD이미지, DVD이미지, RPM파일 등)하여 서비스하고 있으며, 데이터 중복이 50% 이상 발생하고 있다. 이와 같이 저장 시스템에 존재하는 데이터 중복에 대한 효율적인 관리를 위하여 저장 시스템을 설계하는 단계에서 중복 제거를 위한 방안이 제공되어야 할 필요성이 있다. 일부에서는 저장 미디어에 대한 기술 발전으로 하드 디스크 및 플래시 미디어에 대한 단위 공간 대비 가격 비율이 점점 낮아지고 있기 때문에 복잡한 소프트웨어/하드웨어 설계 보다 간단한 설계 방식이 성능 면에서 좋을 것이라고 문제를 제기할 수 있다. 그러나 데이터 중복 제거를 통한 관리는 여러 면에서 이점을 찾을 수 있다. 첫째, 중복된 데이터 제거함으로써 불필요한 저장 공간을 줄일 수 있다. 둘째, 중복된 데이터를 공유함

으로써 캐싱(caching) 효과를 일으켜 성능을 향상시킬 수 있다. 셋째, 모바일 환경에서는 중복데이터의 전송을 줄임으로서 에너지 소모와 네트워크 비용을 절감할 수 있다.[1]

본 논문에서는 이중 시스템 통합을 위한 데이터 관리 기법을 모델링하여 보다 효율적으로 데이터 중복을 제거하기 위한 모델을 제안한다.

2장에서는 데이터 중복 제거와 관련된 최근 연구 동향 및 관련연구에 대해서 알아보고, 3장에서는 제안하는 시스템 통합 관리 방법 중 데이터 중복 제거의 설계 원리와 동작에 대해 설명한다. 4장에서는 제안하는 시스템의 예상 시나리오에 대해 기술하였으며 마지막으로 5장에서는 결론 및 향후 연구방향을 제시한다.

2. 관련연구

2.1. 연구동향

저장 시스템이나 컨트롤 프로그램 등 다양한 시스템에서 중복 데이터를 제거해 네트워크 대역폭과 저장효율을 향상시키려는 연구가 진행되었다.

Chord, Pastiche 같은 P2P 오버레이 네트워크에서 라우팅과 데이터의 중복 저장을 막기 위해 MD5, SHA1 해시 함수를 사용하고, 특히 저장 시 데이터 해시를 만들고 서로 비교하여 같은 해시이면 중복으로 처리하여 저장을 막는다.[2],[3]

잘 알려진 Rsync는 네트워크로 연결된 데이터를 동기화 시켜주는 프로그램이다. 데이터의 삭제나 수정 같은 변화가 생길 경우 데이터 복사가 일어나는데 이때 Rolling Checksum이라는 중복 데이터를 검색하는 알고리즘을 사용해 새로운 데이터의 복사만 일어나게 하는 프로그램이다. Rolling Checksum은 원본 파일을 일정하게 나누어 블록단위의 해시값을 생성하고 비교하려는 파일은 Sliding 기법을 이용하여 바이트 단위별로 모든 블록을 overlapping시켜 해시를 만들고 원본파일의 해시 리스트와 비교하여 중복된 부분을 찾는다. Rsync방식은 모든 중복 데이터를 찾을 수 있어 중복제거 성능에서 우수하지만 수행시간이 길어 대용량 데이터를 처리하는 저장 시스템에서 사용이 어렵다.[4]

Plan9의 Venti는 네트워크 저장 시스템에서 중복 데이터를 제거하여 저장하는 시스템이다. Venti는 데이터를 저장할 경우 데이터를 고정된 크기의 블록으로 나누고 각 블록에 SHA1 해시를 적용하여

160bit 크기의 해시를 만들고 전송한다. 만약 블록의 해시가 저장 서버에 있을 경우 중복으로 간주하여 블록의 저장을 피한다. Venti는 이전 저장 정보가 없이도 델타(delta) 백업의 효과를 낼 수 있으며 실험 결과에서는 다른 스냅샷(snapshot) 시스템과는 달리 30%의 저장공간을 줄이는 것으로 나타났다. Venti의 단점으로는 중복 데이터를 놓치는 경우가 발생하는 것을 들 수 있다.[5],[6]

이외에도 최근의 연구로 Data Domain의 Summary Vector, Stream-Informed Segment Layout 기법 및 HP의 Sparse Indexing 등이 제시되고 있다. 이와 같은 연구는 주로 대용량의 데이터를 처리하는데 있어서 문제가 되고 있는 해시 데이터를 효율적으로 처리하는데 초점을 맞추고 있다.[7],[8]

2.2. Minimal Cost Spanning Tree

각 가지의 비용이나 시간을 나타내는 가중치를 주어진 그래프에서 순환없이 모든 정점을 연결하여 가중치가 가장 적게 드는 그래프를 Minimal Cost Spanning Tree 혹은 Minimal Spanning Tree라 한다. 가중치 그래프에는 적어도 1개 이상의 Minimal Cost Spanning Tree가 존재한다.

Minimal Cost Spanning Tree는 다음과 같은 성질을 가진다.

- 1) 그래프 내에 원래 있던 edge들만을 포함한다.
- 2) vertex의 수가 n이면, n-1개의 edge만을 포함한다.
- 3) 순환(cycle)이 있어서는 안 된다.

이러한 Minimal Cost Spanning Tree는 컴퓨터 네트워크 설계나 수송시스템에 주로 이용된다. 예를 들어 어떠한 거대 네트워크가 있는데 어느 곳에 컴퓨터를 연결하든지 다른 모든 컴퓨터와 상호 통신이 가능하도록 만들고자 한다. 각 네트워크는 소규모 네트워크의 집합이고 네트워크와 네트워크의 사이에는 라우터가 통역을 담당하고 있다. 라우터와 라우터간에 Ping Time이 짧은 것도 있고 긴 것도 있다고 한다면, 라우터들이 최소한의 연결만을 가지면서도 컴퓨터들이 가장 빠르게 통신할 수 있는 경로를 찾으려 할 때 쓰인다. 마찬가지로 수송시스템의 경우를 살펴보면, 모든 도시를 연결할 때 모든 도시마다 1:1로 길을 내면 가장 빠르겠지만 막대한 고속도로 건설비가 들어가므로 매우 비효율적일 것이다.

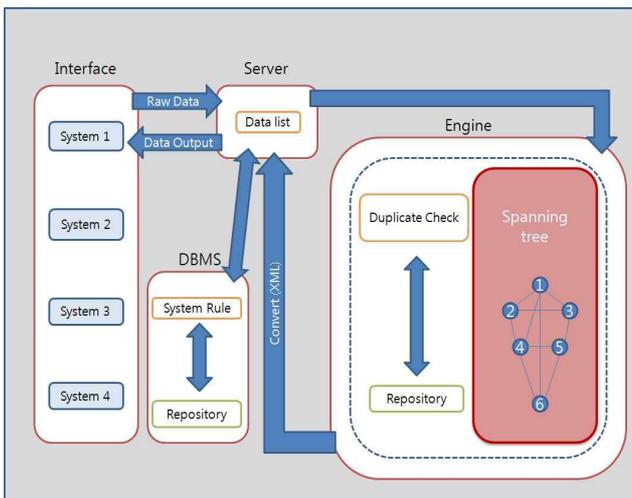
따라서 예산을 최대한 적게 쓰면서도 모든 도시가 가장 효율적으로 교통 통행이 가능하게 하려면 어떤 도시들 사이에 고속도로를 내어야 하는가에 대한 문제를 해결할 때도 쓰인다.[9]

3. 데이터 관리 방법 제안

본 논문에서는 분산된 시스템 상의 데이터를 검사, 관리하는 시스템을 두어 각 사용자들이 스스로 시스템을 일관된 방법으로 통합하지 않아도 각 데이터의 관리가 가능하도록 할 수 있는 방법을 제안한다. 그림.1은 본 연구에서 제안하는 통합 시스템의 전체적인 흐름도이다.

전체 시스템은 크게 5가지로 구성된다.

- System : 이종 시스템의 각 인터페이스.
- Server : 데이터 흐름 제어 및 데이터에 대한 추가, 업데이트 이벤트 발생 시 저장 후 시스템 규칙에 따라 DBMS에 저장.
- Engine : Minimal Cost Spanning Tree의 원리에 따라 신속하고 효율적으로 중복여부 검사, 저장하여 XML로 변환.
- DBMS : 별도의 저장소를 두어 사용자에게 최근 업데이트된 데이터를 분류.
- XML 컨버터 : 데이터를 XML화합으로써 속성 및 구조 일관성 유지 가능.



[그림 1] System Layout

본 시스템은 분산된 시스템의 공통된 데이터들에 관한 리스트를 생성하는 단계와 이를 토대로 검사 및 관리하는 단계로 구분된다.

리스트를 생성하는 단계는 각 이질적인 데이터의 메타데이터에 대한 DBMS에서 정의된 Rule을 기반으로 종류 및 형식에 따라 메타데이터로 자동 생성이 된다. 리스트는 메타데이터를 일관된 형식으로 변환시켜 비교를 용이하게 할 뿐만 아니라, 각 데이터에 대한 구조를 유지할 수 있다는 측면에서 사용이 된다. 메타데이터는 대량의 정보 중 찾고 있는 정보를 효율적으로 검색하고 이용하기 위해 일관된 형태로 데이터에 대한 정보를 주요 내용으로 하는 데이터이다.

검사 단계는 임의의 시스템의 메타데이터를 입력받게 되면 DBMS의 데이터 비교를 위한 임시 저장된 리스트와 함께 Minimal Cost Spanning Tree의 원리에 따라 검사하게 되는데, 검색빈도가 높거나 최신 데이터에 대해 낮은 가중치를 두어 체크하도록 하여 향후 타 트리구조 기반의 빠른 검색을 통한 향상된 성능으로 중복된 데이터를 비교하고 제거할 수 있다. 처리된 데이터는 XML 포맷으로 변환하여 서비스 환경에서 시스템과 DBMS로 전송되도록 한다. 그림.2는 정리된 데이터의 XML구조를 보여주기 위한 예이다.(데이터는 본 논문과 상관없음.)

```

<?xml version="1.0" encoding="euc-kr" ?>
<?XML-stYLESHEET type="text/xml"?>
- <logilist>
  <!-- A movieman information -->
  - <L1>
    <id>s128792</id>
    <name>Dong-Gun, Kang</name>
    <socialid>720307-*****</socialid>
    <company>AM ent.</company>
    <work>broadcast?3,movie?17,album?4,etc?8</work>
    <update>100502</update>
  </L1>
  - <L2>
    <id>k378943</id>
    <name>In-Sun, Ko</name>
    <socialid>871006-*****</socialid>
    <company>The poy ent.</company>
    <work>broadcast?2,movie?12,etc?9</work>
    <update>100503</update>
  </L2>
</logilist>

```

[그림 2] Structure of XML

4. 시나리오

본 시스템은 다음과 같은 시나리오로 실제 상황에서 프로세스가 진행된다.

시스템 초기 시 시스템 A의 사용자가 Raw-data를 서버에 전송하면 서버는 DBMS에 있는 Rule에

맞게 리스트화하여 검사 엔진으로 보내진다. 데이터는 엔진 내 저장공간에 있는 임시 리스트와 비교되며, 이 때 Minimal Cost Spanning Tree의 원리를 적용하여 비교를 실시하게 된다. 그래프의 운행 방식인 DFS나 BFS 중 어느 방식이 사용되더라도 간선들의 집합인 T의 연결선들은 신장 트리를 형성하게 되는데, DFS 운행 알고리즘을 이용하여 만들어진 신장 트리를 깊이 우선 신장 트리(depth-first spanning tree)라 하고, BFS 운행 알고리즘을 이용하여 만들어진 신장 트리를 넓이 우선 신장 트리(breadth-first spanning tree)라 한다. 이 방법을 통해 보다 빠르고 정확하게 데이터 비교를 실시하고, 중복된 데이터를 발견하여 제거할 수 있다. 중복된 데이터가 없다면 XML로 변환되어 시스템 사용자와 DBMS에 전송되어 각각 결과물 출력 및 저장이 이루어진다.

시스템 B의 사용자가 Raw-data를 서버에 전송하면 서버는 DBMS에 있는 Rule에 맞게 리스트화하여 검사 엔진으로 보내진다. 데이터는 엔진 내 저장공간에 있는 임시 리스트와 비교되며, 이 때 Minimal Cost Spanning Tree의 원리를 적용하여 비교를 실시하고 중복된 데이터를 발견하여 제거할 수 있다. 중복된 데이터가 없다면 이 또한 XML로 변환되어 시스템 사용자와 DBMS에 전송되어 각각 결과물 출력 및 저장이 이루어지겠지만, 중복된 데이터가 있다면 시스템 사용자인 B뿐만 아니라 이전 데이터 입력 시스템의 사용자에게 갱신여부를 메시지로 통보하여 데이터의 변경여부를 확인할 수 있게 한다.

위와 같은 두 가지의 시나리오를 통해 각 시스템들은 사용자가 어떠한 시스템에서 접속해도 갱신된 최신의 데이터를 확인해 볼 수 있다. 위에서 제시한 중복 데이터 제거에 대한 방법을 통해 검사 및 확인에서 소비되는 시간을 줄여 프로세스 효율성 및 데이터 무결성을 향상시킬 수 있다. 또한 지속적인 데이터 관리 및 갱신을 통해 이중 시스템들 간에 보다 효과적인 서비스를 제공할 수 있다.

5. 결론 및 향후연구

본 연구에서는 이질적인 데이터 위주의 시스템 통합 간의 데이터 무결성과 활용 효율성을 향상시키기 위해 Minimal cost spanning tree의 원리를 적용한 시스템 구조를 제시하였다. 제안된 시스템은 수행시간이 길어 대용량 데이터를 처리하는 저장 시스템에

서 발생하는 작업시간을 감소시키고 이중시스템 간의 데이터 교류 및 적응력이 높은 연계시스템을 기대할 수 있다.

향후 연구로는 통합 인증 시스템에 결합된 데이터의 유효성을 검증하고, 데이터 정보의 안정성과 시스템 확장을 보장하기 위한 연구가 필요하다.

참고문헌

- [1] Ho Min JUNG, Young Woong KO, "Storage System Performance Enhancement Using Duplicated Data Management Scheme", 정보과학회논문지 : 시스템 및 이론 제37권 제1호, 2월, 2010.
- [2] J.S. Robin, C.E. Irvine. "Analysis of the Intel Pentium's ability to support a secure virtual machine monitor", In Proceedings of the 9th USENIX Security Symposium, Denver, CO, August 2000.
- [3] KyoungSoo Park, Sunghwan Ihm, Mic Bowman, and Vivek S. Pai., "Supporting Practical Content-Addressable Caching with CZIP Compression", In Proceedings of the USENIX Annual Technical Conference, Santa Clara, CA, June 2007.
- [4] A. Tridgell. "Efficient algorithms for sorting and synchronization", The Australian National University, 1999.
- [5] <http://plan9.bell-labs.com/plan9/>
- [6] QUINLAN, S., AND DORWARD, S. "Venti: a new approach to archival storage", In Proceedings of the 1st USENIX Conference on
- [7] B. Zhu, K. Li, and H. Patterson, "Avoiding the disk bottleneck in the data domain deduplication file system," in Proceedings of the Seventh USENIX Conference on File and Storage Technologies (FAST), pp.269-282, 2008.
- [8] Mark Lillibridge, "Sparse Indexing, Large Scale, Inline Deduplication Using Sampling and Locality," In Proceedings of the Seventh USENIX Conference on File and Storage Technologies (FAST) 2009, San Francisco, CA.
- [9] Chonbuk National University , Internet Lab