2002년도 대한전자공학회/대한전기학회 시스템 및 제어분야 합동 추계학술대회 논문집(2002.11.30)

분산 FTP 서버의 ACE 기반 로그 마이닝 시스템

민 수 홍, 조 동 섭
이화여자대학교 과학기술대학원 컴퓨터학과
전화 : 02-3277-2309 / 핸드폰 : 016-358-7118

Distributed FTP Server for Log Mining System on ACE

Su Hong Min, Dong Sub Cho
Dept. of Computer Science and Engineering, Ewha Womans University
E-mail : shmin@ewha.ac.kr, dscho@ewha.ac.kr

Abstract

Today large corporations are constructing distributed server environment. Many corporations are respectively operating Web server, FTP server, Mail server and DB server on heterogeneous operation. However, there is the problem that a manager must manage each server individually. In this paper, we present distributed FTP server for log mining system on ACE. Proposed log mining system is based upon ACE (Adaptive Communication Environment) framework and data mining techniques. This system provides a united operation with distributed FTP server.

I. 서론

기업의 전산 환경은 분산 시스템 형태로 변화되고 있다. 과거와 달리 인터넷 이용이 증가하면서 대다수의 기업은 웹서버, 파일 서버, 메일 서버, 데이터베이스 서버 등을 각각의 하드웨어에 분산해서 운영하고 있다. 이러한 환경에서 시스템 관리자는 데이터 백업과 같은 반복적인 관리 작업 외에 다수의 서버가 잘 동작하고 있는지 항상 주지해야 하며, 주기적으로 각 서버별로 PING을 보내 서버의 응답 유무를 검사해야 한다. 또한 각각의 분산된 서버들을 개별적으로 관리해야 하는 문제점이 있다. 따라서 본 논문에서는 이러한 분산 시스템 환경의 문제점을 해결하기 위한 방안으로 로그 마이닝 시스템을 제안하고자 한다. ACE 기반 로그 마이닝 시스템은 ACE 플레임웍과 데이터마이닝 기법을 이용해 설계하였으며, 분산되어 있는 각각의 FTP 서버의 로그 정보를 이용해 서버별 이용률, 사용자 정보, 구동 유무를 통합해서 효율적으로 관리 운영할 수 있도록 하는데 목표가 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 ACE (Adaptive Communication Envrionment)와 데이터마이닝 기법에 대해 기술하고, 3장에서는 본 논문에서 제안하는 ACE 기반 로그 마이닝 시스템을 설계한다. 4장에서는 결론 및 향후 방향에 대해 기술한다.

이 논문은 2002년도 두뇌한국21사업에 의하여 지원되었음

II. 관련 연구

2.1 ACE (Adaptive Communication Environment)

ACE는 병행성 (concurrent) 통신 소프트웨어의 핵심 패턴들을 구현한 오픈 소스기반의 객체지향형 프레임워크이다. ACE는 크로스 플랫폼 기반의 재사용 가능한 C++ wrapper facade와 일반적인 통신 소프트웨어 상에서 실행되는 프레임워크 요소들의 집합이다. ACE

에 의해 제공되는 통신 소프트웨어 기능들은 다음과 같다 - 이벤트 디멀티플렉싱, 이벤트 핸들러 디스패칭, 시그널 핸들링, 서비스 초기화, 프로세스간 통신, 공유메모리 관리, 메세지 라우팅, 분산 서비스들에 대한 동적 (재)설정, 병행성과 동기화 등이 있다.

ACE는 고성능 실시간 통신 서비스와 애플리케이션을 개발을 목표로 한다. ACE를 사용하면 프로세스간 통신, 이벤트 디멀티플렉싱, 명시적 동적 링킹, 병행성 객체지향 네트워크 애플리케이션과 서비스의 개발을 단순화시킬 수 있다. 추가적으로 ACE는 시스템 설정과 동적으로 연결된 서비스들에 의한 실시간 재설정, 프로세스와 쓰레드상에서 서비스들을 실행하는 것을 자동화해준다. ACE를 다음과 같은 이점이 있다.

· 이식성의 증가 - ACE 컴퍼넌트들은 하나의 운영체제상에서 병행성 네트워크 애플리케이션들을 구현하기 쉽게 해주며, 수많은 다른 운영체제로 그것을 포팅하는 작업을 손쉽게 해준다.

· 소프트웨어 품질의 증가 - ACE 컴퍼넌트는 통신 소프트웨어의 핵심 품질요소들을 증진시키기 위해 많은 핵심 패턴을 사용해서 디자인되었다.

· 효율과 예측력(predictability)의 증가 - ACE는 넓은 분야의 애플리케이션 QoS 요구를 지원하기 위해 디자인되었다. 이런 요구에는 딜레이(delay)에 민감한 애플리케이션을 위한 낮은 지연(latency), 대역폭에 민감한 애플리케이션을 위한 높은 성능, 실시간 어플리케이션을 위한 예측 능력 등이 있다 [1].

2.2 데이터마이닝 (Data Mining) 기법

데이터마이닝이란 대량의 데이터로부터 쉽게 드러나지 않는 유용한 정보들을 추출하는 과정을 말한다. 여기서 정보는 묵시적이고 잘 알려져 있지 않지만 잠재적으로 활용가치가 있는 정보를 말한다. 다시 말해 기업이 보유하고 있는 일일 거래자료, 고객자료, 상품자료, 마케팅 활동의 피드백 자료와 기타 외부자료를 포함하여 사용 가능한 데이터를 기반으로 숨겨진 지식, 기대하지 못했던 패턴, 새로운 법칙과 관계를 발견하고 이를 실제 경영의 의사결정 등을 위한 정보로 활용하고자 하는 것이다. 정보를 찾아내는 방법은 어떤 특정 기법과 그 기술 자체만을 의미하는 것이 아니고, 비즈니스 문제를 이해하고 이러한 문제를 해결하기 위하여 정보기술을 적용하는 포괄적인 과정을 의미한다. 따라서 데이터마이닝을 효율적으로 수행하기 위하여 시계열분석 등 각종 통계기법과 데이터베이스 기술뿐만 아니라 산업공학, 신경망, 인공지능, 전문가시스템, 퍼지논리, 패턴인식, 기계적 학습, 불확실성 추론 (Reasonging with Incertainty), 정보검색에 이르기까지 각종 정보기술과 기법들을 사용한다 [6].

III. ACE 기반 로그 마이닝 시스템

3.1 시스템 개요

본 논문에서 제시하는 로그 마이닝 시스템은 사용자가 분산되어 있는 FTP 서버의 로그 정보와 상태 정보를 읽어와 FTP 서버를 효율적으로 관리하는 데 그 목적이 있다. 다음은 분산 FTP 서버의 ACE 기반 로그 마이닝 시스템의 전체 구조이다.

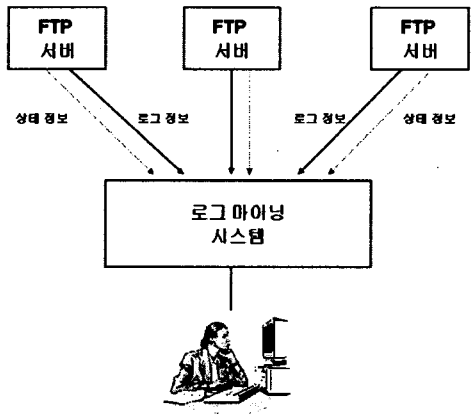

그림 1 ACE 기반 로그 마이닝 시스템

분산되어 있는 각각의 FTP 서버에는 클라이언트에 대한 로그 정보가 생성된다. FTP 서버에서 생성된 로그 정보는 주기적으로 로그 마이닝 시스템의 메시지 큐로 전송된다. 로그 마이닝 시스템의 메시지 큐는 분산되어 있는 FTP 서버의 로그 정보들을 수신한 후, 이를 데이터베이스 서버로 전송한다. 데이터베이스 서버에 저장된 로그 정보들은 데이터마이닝 툴을 이용해 사용자가 원하는 의미 있는 데이터로 생성되며, 이때 생성된 데이터는 사용자 요청에 의해 웹 서비스를 이용해 전달된다. 본 논문에서 제시한 로그 마이닝 시스템은 각각의 FTP 서버의 이용률을 높일 수 있으며, 로그 정보를 통해 FTP 서버를 이용하는 클라이언트에 대한 정보를 얻을 수 있다. 또한 여러 대의 분산된 FTP 서버를 통합해 효율적으로 운영, 관리할 수 있다.

3.2 로그 마이닝 시스템의 설계

(1) 시스템 환경

- DB Server: MS SQL Server 2000
- Data Mining tool: MS SQL Analysis Service
- OS: MS Windows 2000 Advance Server

- Development tool: ACE 5.2 toolkit
- Language: C++

(2) 시스템 설계

로그 마이닝 시스템은 FTP 서버로부터 로그 정보를 읽어 들이는 메시지 큐와 저장된 로그 정보를 데이터베이스 서버로부터 추출해서 MS SQL Analysis Service를 통해 데이터마이닝 기법을 적용하는 부분, 분산된 FTP 서버의 상태정보를 점검하는 세 부분으로 나눌 수 있다. 다음은 로그 마이닝 시스템의 구조이다.

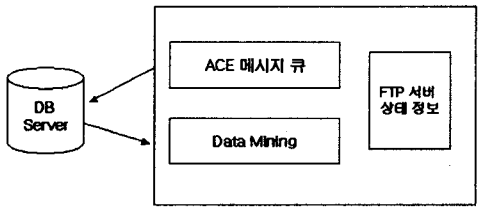

그림 2 로그 마이닝 시스템

▶ ACE 메시지 큐

ACE의 메시지 큐는 Unix System V 메시지 큐를 모델로 만들어졌다. ACE에서 메시지는 메시지 블록의 형태로서 메시지 큐에 적재된다. 각 메시지 블록은 헤더와 데이터 블록을 포함한다. 메시지 블록은 데이터 블록 또는 메시지 헤더의 포인터를 가지며, 데이터 블록은 실제 데이터 버퍼를 가르키는 포인터를 담고 있다. 한 개의 데이터 블록은 두 개의 메시지 블록이 공유할 수 있어 데이터 버퍼를 효율적으로 사용할 수 있다.

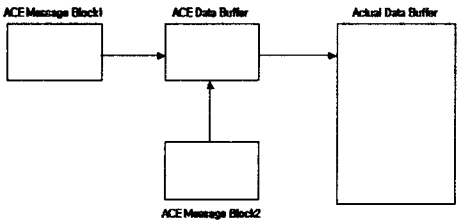

그림 3 다수의 메시지 블록을 이용한 데이터 공유

ACE 메시지 큐를 생성하기 앞서, 메시지 블록을 생성한다. 메시지 블록은 ACE_Message_Block 클래스를 사용해 생성하는데, 이 클래스는 데이터 메모리를 할당하고, 관리하는데 사용된다.

```
ACE_Message_Block (size_t size,
    ACE_Message_Type type = MB_DATA,
    ACE_Message_Block *cont = 0,
    const char *data = 0,
    ACE_Allocator *allocator_strategy = 0,
    ACE_Lock *locking_strategy = 0,
    u_long priority = 0,
    const ACE_Time_Value & execution_time
                = ACE_Time_Value::zero,
    const ACE_Time_Value & deadline_time
                = ACE_Time_Value::max_time);
```

그림 4 ACE Message Block 클래스

ACE_Message_Block은 rd_ptr(), wr_ptr() 메소드를 호출해서 데이터 블록 안으로 데이터를 삽입한다. 메시지 블록 생성이 끝나면, ACE 메시지 큐를 구현한다. 앞에서 생성한 메시지 블록을 메시지 큐에 초기화해서 삽입한다. ACE는 여러 타입의 메시지 큐를 가지는데, 일반적으로 사용하는 정적 메시지 큐와 실시간으로 사용하는 동적 메시지 큐로 나눌 수 있다. 본 논문에서는 ACE 정적 메시지 큐를 이용해 분산된 FTP 서버의 로그 정보를 주기적으로 읽어와 데이터베이스 서버에 저장한다 [4].

```
int start_test(){
        for(int i=0; i<no_msgs_;i++){
    //Create a new message block of data buffer size 1
     ACE_Message_Block * mb=
     new ACE_Message_Block(SIZE_BLOCK);
    //Insert data into the message block using the rd_ptr
            *mb->wr_ptr()=i;
    //Be careful to advance the wr_ptr
            mb->wr_ptr(1);
    //Enqueue the message block onto the message queue
            if(this->mq_->enqueue_prio(mb)==-1){
                ACE_DEBUG((LM_ERROR,"\nCould
                    not enqueue on to mq!!\n"));
                return -1;
            }
            ACE_DEBUG((LM_INFO,"EQ' d data:
                %d\n",*mb->rd_ptr()));
        }
        //Use the iterators to read
        this->read_all();
        //Dequeue all the messages
        this->dequeue_all();
        return 0;
    }
```

그림 5 ACE 메시지 큐 생성

▶ 데이터마이닝 기법

본 논문에서는 데이터마이닝 기법을 적용해 FTP 서버의 로그 정보를 사용자가 원하는 의미 있는 데이터로 추출하도록 설계하였다. 사용자는 로그 데이터들을 데이터마이닝 기법인 연관 규칙을 적용해 로그 데이터들간의 연관성 정도를 측정할 수 있다. 연관 규칙이란 상품이나 서비스의 거래기록(Historical) 데이터로부터 상품간의 연관성 정도를 측정하여 연관성이 많은 상품들을 그룹화 하는 클러스터링의 일종이다. 연관성 측

정에서 얻어지는 결과물인 연관 규칙은 'If A, then B (A->B)' 와 같은 형식으로 표현되고 '상품 A가 구매되어진 경우는 상품 B도 구매된다' 라고 해석한다 [3]. 본 논문에서는 대량의 FTP 서버의 로그 정보를 MS SQL Analysis Service를 이용해서 시간대별 사용자 수와 파일 전송 크기, 저장된 파일 종류에 따른 서버의 이용률 등의 정보를 추출한다.

▶ FTP 서버의 상태 정보

FTP 서버의 상태 정보 (서버가 구동되는지의 여부)는 ACE 소켓 클래스(ACE_SOCK)를 이용해서 설계하였다. 사용자는 주기적으로 분산되어 있는 FTP 서버들의 상태를 멀티캐스트 방식으로 요청한다. 요청 받은 각각의 FTP 서버는 클라이언트에게 자신의 IP 주소와 상태 정보를 데이터그램(Datagram)으로 전송한다. 다음은 ACE_SOCK 클래스의 하위 클래스들이다.

· ACE_SOCK_Dgram: UDP 데이터그램 프로토콜을 기반으로 하며, sendto() 와 receivefrom() 함수들을 호출한다.

· ACE_SOCK_Acceptor: accept()와 listen() 함수를 이용해 수동적인 연결을 지원한다.

· ACE_SOCK_Connector: connect() 함수를 호출하며, 능동적인 연결을 지원한다.

· ACE_SOCK_Mcast: 데이터그램 기반 멀티캐스트를 지원한다.

각각의 FTP 서버 리스너(listener)는 사용자의 요청을 수신해서, 서버의 IP 주소, 상태 정보를 전송한다.

```
int accept_data() {
    int byte_count = 0;
    while((byte_count = local_.recv(data_buf,
      SIZE_DATA, remote_addr_))!= -1 {
          data_buf[byte_count]=0;
      ACE_OS::sleep(1);
      If(send_data)==-1) break;
   }  return -1
}
int send_data()
{
    ACE_OS :: sprintf(data_buf, "Server says hello
     to you");
    if ( local_.send(data_buf, ACE_OS:: strlen(data_buf)+1,
       remote_addr_)==-1)
          return -1;
   else  return 0;
}
```

그림 6 FTP 서버 리스너

클라이언트는 분산된 FTP 서버에게 동일한 메시지를 요청하기 때문에 ACE_SOCK_Dgram_Mcast 클래스를 이용해 멀티캐스트 방식으로 동작한다 [5].

```
class Sender_Multicast{
public:
sender_Multicast(int port):
    local_addr_((u_short)0), dgram_),
    multicast_addr_(port,
    DEFAULT_MULTICAST_ADDR) {
    }

//메시지를 분산된 FTP 서버로 전송한다.
int send_to_multicast_group()
{
    mcast_info = htons (1000);
    if(dgram_.send (&mcast_info, sizeof(mcast_info,
      multicast_addr_)==-1)
          retrun -1;
   ACE_DEBUG ((LM_DEBUG, "%S; Sent multicast
      to group", _FILE_, mcast_info));
   return 0;
}
private:
   ACE_INET_Addr multicast_addr_;
   ACE_INET_Addr local_addr;
   ACE_SOCK_Dgram_;
   int mcast_info;
};
```

그림 7 로그 마이닝 시스템의 FTP 상태 정보

Ⅳ. 결론

본 논문에서는 ACE 기반 로그 마이닝 시스템을 설계해 분산된 각각의 FTP 서버를 통합해 효율적으로 관리 운영하는 방법을 제시하였다. 우리가 제안한 ACE 기반 로그 마이닝 시스템은 FTP 서버의 로그 정보를 이용해 사용자별 이용현황, 서버별 이용 상태, 서버의 구동 유무를 통합해서 관리할 수 있으며, 데이터마이닝 기법을 도입해 대량의 로그 데이터를 바탕으로 실제 유용한 정보를 추출해 낼 수 있도록 설계하였다.

향후 연구로는 본 논문에서 제안한 ACE 기반 로그 마이닝 시스템을 구현, 검증할 예정이며, 다양한 데이터마이닝 기법을 이용해 로그 분석을 하고자 한다.

참고문헌

[1] Douglas C. Schmidt, Stephen D. Huston, "C++ Network Programming Volume 1 - Mastering Complexity with ACE and Patterns," Addison-Wesley. 2002.

[2] Douglas C. Schmidt, Stephen D. Huston, "C++ Network Programming Volume 1 - Mastering Complexity with ACE and Patterns," Addison-Wesley. 2002.

[3] 조재희, 박성진, "OLAP 테크놀로지", 시그마 지식경영연구서, 2000. 1.

[4] Umar Syyid, "A Tutorial Introduction to the ADAPTIVE Communication Environment (ACE) - Message Queues", HUGHES, Chapter 9

[5] Umar Syyid, "A Tutorial Introduction to the ADAPTIVE Communication Environment (ACE) - IPC SAP", HUGHES, Chapter 2

[6] http://user.chollian.net/~keyman21/cp/mining.html