

# DeVAC 서버 프로그램의 안정성 평가 및 개선

## Evaluation and Improvement of Stability of DeVAC Server Program

박선영

Park Sun-Young

부산대학교 컴퓨터공학과

parksy@pusan.ac.kr

### ABSTRACT

유사 문서 탐색 시스템인 DeVAC(Document eVolution Analysis Center)에서는 사용량이 많은 기관 사용자들의 편리한 사용을 위해 특수 사용자 버전 DeVAC Manager를 운영하고 있다. 특수 사용자용 DeVAC Manager는 DeVAC 서비스를 제공하기 위하여 DeVAC 서버에 Demon Process 형태의 서버 프로그램(dserver)의 동작을 필요로 하는데, 이 프로그램이 비정상적으로 동작하여 로그인이 수행되지 않는 경우가 많이 발생하고 있다. 이 현상이 발생했을 경우 해당 서버 프로세스를 죽인(kill) 후 다시 프로세스를 시작하여야만 정상적으로 DeVAC Manager의 기능을 활용할 수 있다. 본 보고서에서는 dserver의 소스 코드 및 오류 코드를 분석하여 서버 프로그램의 기능적 요소 및 에러 현상의 원인을 밝혀낸다. 또한 밝혀낸 원인을 제거하고 예외 처리를 강화함으로써 서버 프로그램을 더욱 견고(robust)하게 개선한다. 분석 결과, MySQL DBMS와의 TCP/IP 연결 중에는 아무런 질의(query)가 발생하지 않는 시간이 일정 시간 이상으로 길어질 경우 해당 연결(connection)이 무효화되는 현상이 발생하는데, 이를 해결하기 위해서는 1) DBMS 설정을 변경하는 방법, 2) MySQL API를 활용하여 연결 무효 시 연결을 재설정하는 방법, 3) DB를 사용할 때만 연결하는 방법 등이 있다. 실험 결과 1)의 방법도 사용 가능하나, 견고하지 못한 방법이라 할 수 있고, 현재와 같이 간헐적으로 서버 프로그램을 사용하는 경우에는 3)의 방법이 더욱 안정적이다. 하지만 추후에 사용자가 늘어날 것을 고려하면, 1)의 방법과 2)의 방법을 동시에 적용하는 것이 성능과 견고함 측면 모두를 고려하였을 때 가장 적합한 것으로 판단된다. 추후 DeVAC 프로그램의 Phyloview 기능을 분석하고 정상적으로 활용하기 위한 방법을 모색할 계획이다.

KEYWORDS DeVAC Manager, MySQL

## 1 서론

사회 각계각층의 표절 논란[1]이 불거지면서 표절 탐색에 대한 관심이 높아졌다. 이에 문서 유사도 탐색 프로그램인 DeVAC[2]시스템에 대한 사용자들이 관심도 커졌다. DeVAC은 문서 표절에 탁월한 성능을 보여주는 문서 유사도 탐색 프로그램이다. DeVAC 시스템의 client side, 즉 DeVAC Manager는 Web을 기반으로 하는 일반 사용자용과 Windows API를 기반으로 하는 기관 사용자용으로 나뉘는데, 기관 사용자용 DeVAC Manager를 사용하기 위해서는 DeVAC 서버에서 서버 프로그램이 정상적으로 가동하고 있어야 한다.

하지만 현재의 DeVAC Manager의 서버 프로그램은 알 수 없는 원인으로 인하여 사용자 인증 기능이 정상적으로 동작하지 않는 경우가 많이 발생하고 있으며, 이러한 현상이 발생할 경우 서버 프로세스를 완전히 종료(kill)하고 몇 분 후 다시 가동하여야만 정상적으로 기능을 사용할 수 있다. 이러한 현상으로 인해 기관 사용자들이 많은 불편을 겪고 있다. 본 보고서에서는 이러한 현상을 해결하기 위하여 서버 프로그램의 소스 코드와 에러 코드를 분석하여 이 현상을 제거하는 것을 목표로 한다.

## 2 DeVAC Manager 서버 프로그램 분석

### 2.1 소스 코드 및 기능 분석

DeVAC Manager의 서버 사이드의 소스 코드를 분석해 본 결과, 프로그램 시작 후 다음과 같은 과정을 거쳐서 클라이언트와 통신을 수행한다.

1. DBMS 연결 초기화
2. 서버 소켓 생성 및 초기화
3. 클라이언트의 요청 메시지 수신
4. 수신된 메시지에 따라 해당 모듈 호출

즉 서버 프로그램은 항상 mySQL DBMS와 TCP/IP를 통해 연결되어 있으면서 클라이언트 프로그램의 요청을 처리한다. 클라이언트 요청 처리를 위해 사용되는 모듈은 다음과 같다.

1. 사용자 인증(로그인) - DeVAC 홈 페이지와 연동
2. DVC 파일 업로드 및 DeVAS 모듈로 검사 수행
3. 사용자의 최근 검사 내역 제공(다운로드를 위한 파일 경로 포함)
4. 검사 완료 파일 전송

사용자 인증 모듈은 DeVAC 홈 페이지의 DB에서 제공하는 사용자 ID와 비밀번호를 이용하여 올바른 사용자인지를 확인한 후 클라이언트에 메시지를 전달한다. DVC 파일 업로드 모듈은 클라이언트에서 생성한 DVC 파일을 즉시 서버측으로 전송하면 서버는 이를 받은 후 DeVAS 모듈을 호출하여 유사도를 탐색하는 기능이다. 검사 내역 모듈은 해당 사용자의 3일 동안의 검사 내역을 전달하며, 사용자가 완료된 검사 내역의 결과 파일을 다운로드받으려 할 경우 파일 전송 모듈을 통해 검사가 완료된 파일을 전송한다.

### 2.2 에러 분석

위에서 언급한 4개의 모듈 중 서버 프로그램 에러가 발생하는 부분은 사용자 인증 모듈이다. 그 중에서도 특히 인증을 위한 쿼리 수행 중 에러가 발생한다. 에러가 발생하면 사용자 인증이 정상적으로 수행되지 않으며, 사용자 인증 후 메인 프로그램이 실행되는 DeVAC Manager의 특성 상 프로그램 사용 자체가 불가능해진다. 현재로서는 에러 발생 시 서버 프로세스를 종료(kill)한 후 재실행하면 반드시 정상적으로 실행되나, 이 과정에서도 소켓이 바로 닫힐 때까지 기다려야 하기 때문에 3~4분 정도가 소요된다. 일단 재실행 후에는 4~5시간 정도는 정상적으로 수행이 되는 것으로 확인되었으나, 대략 하루 정도 지난 이후에는 또 다시 에러가 발생하였다. 이 문제를 해결하기 위하여, 쿼리 수행 중 발생하는 에러 메시지를 확인한 결과 mySQL 에러 번호는 2006 이었다. 이 번호는 mySQL 연결을 더 이상 사용할 수 없게 되었을 때 발생한다[3]. 즉, mySQL과 서버 프로그램 사이의 연결이 무효화되어

정상적인 처리를 할 수 없게 된 것이다. MySQL C API[4]에 따르면, 일반적으로 DBMS와의 TCP/IP 연결은 일정 시간동안 쿼리가 발생하지 않을 경우 해당 연결을 무효화한다. 기본 설정 값은 8시간이다. 즉 8시간 동안 어떤 사용자도 시스템에 로그인하지 않아서 연결이 무효화된 것이다.

```
[root@DEVAC NDEVAC]# ./dserver 2871 &
[1] 30503
[root@DEVAC NDEVAC]# 2871

[root@DEVAC NDEVAC]# Login Called
LoginGranted Message send
UserInfo Received ID:admin
Query make complete
Query executed
Store result Error. MySQL ErrorNo 2006
Count Checked
Login Denied
```

그림 1. DeVAC 서버 프로그램에 접속이 되지 않을 때 핸들링된 에러 메시지. ErrorNo 2006이며, 이는 해당 연결이 무효화되었음을 의미한다.

### 3 에러 현상 수정 방법

지금까지 조사한 바에 의하면 DeVAC 서버 프로그램의 에러는 장시간동안 쿼리가 발생하지 않음으로 인한 MySQL 연결의 무효화가 그 원인이다. 이를 해결하기 위해서는 다음과 같은 3가지 방법을 생각해 볼 수 있다.

1. DBMS 설정 변경
2. MySQL API를 활용한 연결 무효 시 연결 재설정
3. DBMS 연결을 DB를 사용할 필요가 있을때만 일시적으로 연결

#### 3.1 DBMS 설정 변경

DBMS의 설정을 변경하는 것은 매우 간단하게 이 에러를 피할 수 있는 방법 중 하나이다. 통상적으로 MySQL의 설정 파일은 /etc/my.cnf 파일에 존재한다. 여기에서 wait\_timeout 변수에 초 단위의 값을 설정해 주면 이 시간만큼은 쿼리가 존재하지 않더라도 연결이 무효화되지 않는다. DeVAC 시스템의 /etc/my.cnf 파일을 살펴본 결과, wait\_timeout 변수가 설정되어 있지 않았다. 설정하지 않았을 경우 기본 값은 28800(8시간)이며, 최대 설정 가능 시간은 31536000(1년)이다. 실제로 이 값을 1초로 설정한 후 서버를 시작한 결과 실행 즉시 연결이 무효화되는 것을 확인할 수 있었다. 이 값을 최대값인 31536000으로 설정하면 대부분의 에러 현상을 피할 수 있다. 다만 이 방법은 견고한(robust) 방법이라고 볼 수는 없다. 최대값이 1년으로 매우 길다고는 하지만 timeout 이외의 상황에 의해 연결이 무효화되었을 경우에는 아무런 효과가 없기 때문이다. 추가적인 에러 핸들링이 없다면 서버 프로그램은 견고하다고 볼 수 없다.

```

[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
# Default to using old password format for compatibility with mysql 3.x
# clients (those using the mysqlclient10 compatibility package).
old_passwords=1
wait_timeout = 31536000

max_allowed_packet = 64M

[mysqld_safe]
log-error=/var/log/mysql.log
pid-file=/var/run/mysqld/mysqld.pid
~
~
~
~
~
~
~
-- INSERT --

```

그림 2. /etc/my.cnf 파일의 내용. wait\_timeout 변수를 선언하고 최대값인 31536000을 설정해 주었다. 이 변수가 없을 경우 28800(8시간)이 자동으로 설정된다.

### 3.2 MySQL API를 활용한 연결 재설정

주기적으로 연결의 유효성을 검사하여, 유효하지 않을 경우 연결을 재설정하는 방법도 있다. MySQL C API[4]에서 제공하는 int mysql\_ping(MYSQL \*mysql) 함수를 사용하여 구현할 수 있다. 이 함수는 해당 MySQL 연결이 유효한지를 검사하여 연결이 끊어진 경우 다시 연결을 시도하는 함수이다. 다만 이 방법은 사용자가 많을 때 자주 수행하게 되면 서버 프로세스의 성능이 떨어진다는 단점이 있다. 따라서 로그인 직전 등 호출 시점을 적절히 제어하면서 사용하면 높은 안정성을 기대할 수 있다.

### 3.3 DB 사용시에만 일시적으로 연결

DB 연결과 관련한 문제점이 발생할 가능성을 원천적으로 차단하는 방법이다. 클라이언트로부터 요청이 발생하면 서버 프로세스가 그 때 DBMS와 연결을 시작하도록 하여, 해당 요청이 종료되면 DBMS와의 연결도 종료하는 방법이다. 이 방법을 사용하면 예기치 않은 연결 종료 발생 가능성이 매우 낮다. 다만 이 방법은 매 요청마다 DBMS와 연결 및 종료하는 시간이 추가로 소요되므로 성능이 매우 떨어진다. 현재의 DeVAC 시스템은 많은 사용자 요청이 발생하지 않으므로 이 방법이 가장 확실한 방법이지만, 추후 시스템을 확장하여 일반 사용자들도 DeVAC Manager의 로그인 기능을 사용하게 한다고 하면 이 방법을 사용하는 것이 좋지 않을 수 있다.

## 4 실험

### 4.1 실험 방법

기존의 서버 프로세스와 위 절에서 제시한 세 가지 방법을 수행한 경우에 대하여 시간 t1~4에 대해 정상 작동 여부와 100개의 select 쿼리에 대한 수행 시간을 측정하여 비교하였다. 각 시간에 대한 설명은 다음과 같으며 최대 5분의 오차가 존재한다. 프로세스 시작 시간부터 검사 시점까지 쿼리는 전송되지 않았다.

1. 프로세스 시작 직후
2. 프로세스 시작 7시간 후
3. 프로세스 시작 9시간 후
4. 프로세스 시작 24시간 후

위의 시간에 대해 각 100개의 SQL 쿼리를 수행하여 그 시간을 측정하였다. 만약 에러가 발생할 경우 시간 대신 ERROR를 기록하였다.

### 4.2 실험 결과

실험 결과는 표 1 과 같다.

프로세스 시작 후	기존 방법	DBMS 설정 변경	무효시 연결 재설정	일시적 연결
0시간 경과	0.05	0.07	0.09	0.44
7시간 경과	0.07	0.06	0.08	0.47
9시간 경과	Error	0.06	0.11	0.43
24시간 경과	Error	0.05	0.08	0.51

표 1. 각 시간에 대해 기존 방법과 제시한 3가지 방법을 사용한 결과. 기존 방법은 쿼리가 존재하지 않은 상태에서 8시간 경과 후 오류가 발생하는 반면, 제시한 방법들은 모두 정상적으로 동작하는 것을 확인할 수 있다. DBMS 설정 변경의 경우 기존 방법과 거의 같은 연산 시간 내에 쿼리가 모두 수행되고, 무효 시 연결 재설정 방법은 10% 정도 지연이 발생하였으며, 일시적 연결의 경우에는 매 쿼리마다 MySQL 연결을 새로 수행하므로 속도가 느린 것을 알 수 있다.

## 5 결론 및 추후 연구

본 보고서에서는 DeVAC Manager의 서버 프로그램의 오류를 분석하고 이를 정정하기 위한 방법을 모색하였다. 서버 프로그램에서 발견한 오류의 종류는 MySQL Error No. 2006 으로, 프로세스 내에서 해당 연결이 무효화되어 발생한 것이다. 즉 프로세스에서 쿼리를 일정 시간 이상 실행하지 않아서

발생한 것인데, 이러한 현상을 수정하기 위하여 1) mySQL 설정을 변경하는 방법, 2) mySQL API를 활용한 연결 재설정 방법, 3) DB 사용시에만 일시적으로 연결하는 방법 등을 사용하여 실험을 진행하였다. 실험 결과 세 방법 모두 해당 에러를 잡아낼 수 있었으나, 1)의 방법은 timeout이 아닌 다른 이유에 의해서 연결이 무효화될 경우 해당 에러를 핸들링 할 수 없고, 2)의 방법은 연결 유효 상태를 검사하는 것에 계산 시간을 소모하며, 3)의 방법은 사용자의 연결이 많을 경우 성능이 크게 하락하는 단점이 있었다. 결과적으로 현재의 시스템에서는 3)의 방법을 사용하는 것이 안정성 측면에서 가장 뛰어나지만 추후 시스템의 확장을 고려한다면 성능의 하락을 피할 수 없으므로 최종적으로는 1)의 방법을 사용하고 2)의 방법을 로그인 등 일부 구간에서 적절히 병행하여 사용하는 것이 가장 효율적이라고 판단된다. 추후 DeVAC Manager의 Phyloview 기능에 대해 분석하고, 이 기능을 정상적으로 활용하기 위한 방법을 모색할 계획이다.

### 참고 문헌

1. 이기창 이재화, "서울대, 논문 이중게재 논란 원천봉쇄," *MKdotcom*(<http://news.mk.co.kr/>), 2010.
2. 조환규 류창건, 김형준, "Devac(document evolution analysis center)," <http://devac.cs.pusan.ac.kr/>, 2010.
3. pasukun, "mysql wait\_timeout 설정법," <http://blog.naver.com/pasukun?Redirect=Log&logNo=100065023858>, 2010.
4. mySQL, "21.9.3. c api function descriptions," <http://dev.mysql.com/doc/refman/5.5/en/c-api-functions.html>, 2010.