

DeVAC 시스템의 Component 분할을 통한 성능개선

Perfomance Improvement for DeVAC System by Dividing Components

김형준

Kim Hyong-Jun

부산대학교 컴퓨터공학과

hjkim83@pusan.ac.kr

ABSTRACT

표절 탐색 시스템으로서 DeVAC의 성능이 많은 사람들에게 알려지면서 사용자들에게서 많은 새로운 기능들을 요구받게 되었다. 하지만 기존의 DeVAC 시스템은 빠른 성능을 위하여 차후 유지보수와 업그레이드와 관련되어 어떠한 기능도 추가되어 있지 않다. 특히 잘못된 모듈화로 인하여 새로운 서비스를 빠르게 제공하기 힘들기 때문에 모듈화를 통한 새로운 시스템이 필요하게 되었다. 본 보고서에서는 DeVAC에서 제공하거나 제공할 예정인 서비스들을 분석하고 기존의 서비스들과 새로운 서비스들에 모두 적합한 DeVAC 시스템의 모듈화를 보여준다. 그리고 각각의 모듈들이 제공하는 서비스와 각 모듈들의 목적을 명시하여 차후 유지보수에 도움이 되도록 한다. 또한 기존의 시스템과의 성능 비교를 통하여 모듈화를 통한 시스템의 성능 변화를 살펴본다.

KEYWORDS Plagiarism, Dividing Component, Modular Programming

1 Motive

표절 탐색 시스템으로서 DeVAC이 널리 알려지면서 많은 사용자들이 이용을 하고 있다. 하지만 늘어나는 사용자들과 이들의 요구를 만족 시키기에는 현재의 DeVAC 시스템은 한계가 있다. 특히 기존의 DeVAC 시스템은 빠른 성능에 초점이 맞추어져 있었으며 시스템의 개선 또한 성능 위주로 이루어졌다[1]. 다른 서비스적 영역에 대해서는 전혀 고려가 되어있지 않았기 때문에 사용자들의 서비스적 요구에 빠르게 대처하지 못하고 있다. 이는 초기 DeVAC 시스템을 설계할때 현재의 상황을 생각하지 못하였기 때문이며 기존의 시스템을 유지하면서 서비스를 추가하기에는 한계에 이르렀다. 또한 차후 유지보수에도 문제가 될 수 있기 때문에 새로운 시스템 구조의 확립 및 시스템 모듈화가 시급한 문제로 부각되고 있다.

본 보고서에서는 현재 DeVAC 시스템의 상황과 기존 시스템의 문제점에 대해 우선 살펴본다. 특히 성능을 제외한 서비스로서의 DeVAC 시스템의 문제를 살펴보고 현재 DeVAC 시스템의 잠재적인 위험도 살펴본다. 또한 새로 요구되는 서비스들과 요구될 것으로 추정되는 서비스들을 분류하고 이들을 지원하기 위한 시스템 구조를 제안한다. 또한 제안한 시스템 구조를 바탕으로 DeVAC 시스템을 개량하고 모듈화한 다음 시스템의 성능을 측정하여 기존의 시스템과 비교한다. 그 결과 모듈화를 통하여 새로 제안된 시스템 또한 이전의 성능을 우선으로 제작한 시스템에 비교하여 성능이 5% 정도만

하락하는 것을 확인할 수 있었다. 하지만 성능의 하강에도 불구하고 새로 제안된 시스템의 확장성과 안정성은 성능하락을 감수할 가치가 있다는 것을 확인하였다.

2 Preliminary

표절 탐색 시스템으로서 DeVAC은 이제 많은 사람들에게 알려져서 많은 분야에서 사용되고 있다. 현재 DeVAC 시스템에는 200 이상의 사용자가 등록을 하여 사용을 하고 있으며, 월 평균 120 개 이상의 표절 검사가 이루어지고 있다.

DeVAC 시스템의 향상을 위해서 먼저 기존의 DeVAC 시스템의 구조를 분석하고 기존 시스템의 문제점을 분석해야 한다. DeVAC 시스템은 초기버전 이후 현재까지 상당한 업데이트를 거쳐 왔다. 특히 서버 이동을 통하여 웹서비스를 변경[2] 현재의 시스템을 구축하였다.

신규 서버의 구축을 통하여 DeVAC 시스템은 한차례 시스템 구조를 변경하였다. 이를 통하여 기존의 DEVAS 시스템에 웹서비스를 추가하여 현재의 표절 탐색 시스템을 제공하고 있다.

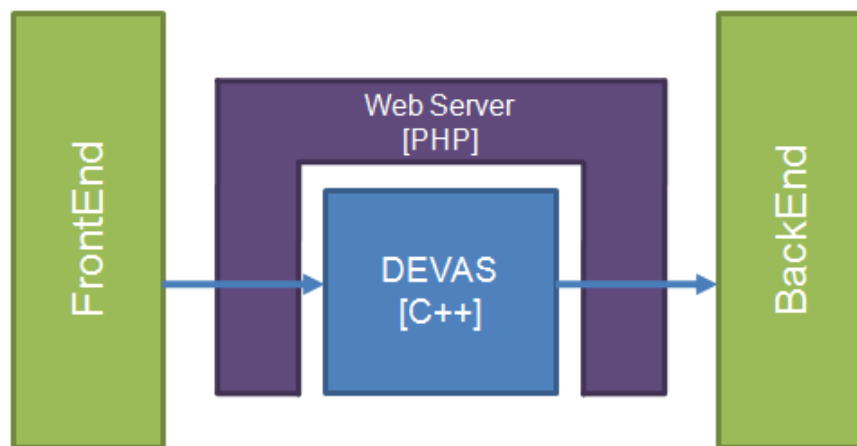


그림 1. 현재 DeVAC 시스템의 구조. 스탠드얼론 시스템인 DEVAS에 웹서버를 이용하여 서비스를 제공한다. 이는 웹서버와 DEVAS 간의 동기화 문제를 초래하고 있다.

그림 1은 기존 DeVAC 시스템의 구조를 나타내고 있다. 프론트엔드에서 전송받은 표절 검사 파일을 웹 서버에서 전송받아 DEVAS 프로그램에 알맞는 구조로 프로세싱한다. 그리고 DEVAS 시스템에서 파일을 검사하고 그 결과를 웹서버에서 다시 BackEnd에 알맞은 구조로 변경하여 BackEnd로 전송하는 방식을 취하고 있다.

우선 이방식은 사용자의 수동 조작이 많이 필요하여 사용법이 복잡하게 되었다. 사용자는 우선 FrontEnd 시스템을 이용하여 표절 탐색 파일을 생성하고 웹서버로 접속한 다음, 표절 탐색 파일을 업로드 해야 한다. 그리고 DEVAS 시스템을 거쳐 표절 탐색이 완료될때까지 기다린 다음 표절 탐색

이 완료되면 웹서버를 거쳐 다운로드 받은 다음 BackEnd에서 검사 결과를 확인하고 있다. 사용자의 수동 조작이 많아지면 자신이 원하는 결과를 정확하게 고를수 있다는 장점이 있지만 이 시스템에서는 번거로운 작업만 추가하는 단점이 되고 있다.

두번째로 웹서버와 DEVAS의 동기화 문제도 있다. 웹서버는 기본적으로 장시간 데이터 전송이 없으면 그 세션을 종료하게 된다. 하지만 대규모 표절 검사의 경우 DEVAS 프로그램에서도 1시간 이상 걸릴 경우도 있다. 이 경우 웹서버 세션이 종료되어 결과 파일을 받아볼 수 없는 치명적인 단점이 발견되었다. 또한 1시간 미만이라고 하더라도 결과가 나올때까지 사용자는 웹 화면만 쳐다봐야 된다는 단점도 존재하였다. 이는 최근에 DEVAS 시스템과 웹서버간의 통신을 비동기적으로 처리함으로써 시간이 오래 걸리는 표절 탐색의 경우 차후 표절 탐색 결과 리스트 화면에서 확인하는 방식으로 처리하여 단점을 제거하였다. 하지만 여전히 사용자가 웹서버를 통하여 결과가 완료되었는지 확인해야 한다는 단점이 존재하고 있다.

또 한가지 DeVAC 시스템의 문제는 DEVAS 프로그램에 있다. DEVAS 프로그램은 현재 파일리스트와 사전파일들을 읽어들여서 표절 탐색 결과를 만들어내는 프로그램이다. 하지만 이 프로그램은 N:N 표절 검사만 지원하고 있기 때문에 1:1 표절 탐색 또는 1:N 표절 탐색은 지원하지 않는다는 단점이 있다. 이로 인하여 표절 탐색 시간이 길어질 뿐만 아니라 새로운 표절 탐색 서비스를 제공하기가 용이하지 않다는 단점이 있다. 그리고 DEVAS 내부 구조가 파일 입출력부터 표절 검사까지 모두 클래스 하나로 묶여져 있기 때문에 차후 유지보수 및 업그레이드도 용이하지 않다는 단점이 존재하고 있다.

마지막으로 웹서버에서 필요시마다 DEVAS를 실행하는 방식으로 말미암아 차후 사용자가 증가하여 동시 표절 검사가 많아질 경우 프로세서가 많이 발생하여 시스템 성능에 악영향을 줄 수 있다. 이는 프로세서를 관리하는 매니저프로그램의 부재로 인하여 발생하는 문제이고 차후 충분히 DeVAC 시스템에 위협이 될 수 있는 문제점이다.

본 보고서에서 제안하는 방식은 프로그램 표절 탐색 시스템인 SoVAC에서 사용된 모델을 이용하고 있다[3]. 또한 기존의 DeVAC 시스템에서의 문제점을 해결하기 위하여 웹서버와 DEVAS에 초점을 맞추어서 모듈화를 진행한다. 특히 DEVAS 시스템을 기능별로 모듈화함으로써 차후 유지보수 및 새로운 서비스에 빠르게 대응하고자 한다.

3 DeVAC 모듈화

DeVAC 시스템의 모듈화를 위하여 우선 DeVAC 시스템에서 제공하고 있는 기능을 분류해야 한다. DeVAC 시스템에서 제공하고 있는 주요 기능은 다음과 같다.

- 표절탐색 파일 제작.

- 표절탐색 파일 업로드.
- 표절탐색.
- 표절탐색 결과 다운로드.
- 표절탐색 결과 확인.

현재 표절탐색 파일 제작 및 결과 확인은 DevacMan에서 제공하고 있다. 그리고 파일 업로드와 다운로드는 사용자의 수동 조작이 필요로 하고 있는 부분인데 DeVAC시스템을 개량하면서 업로드와 다운로드도 DevacMan에 통합한다.

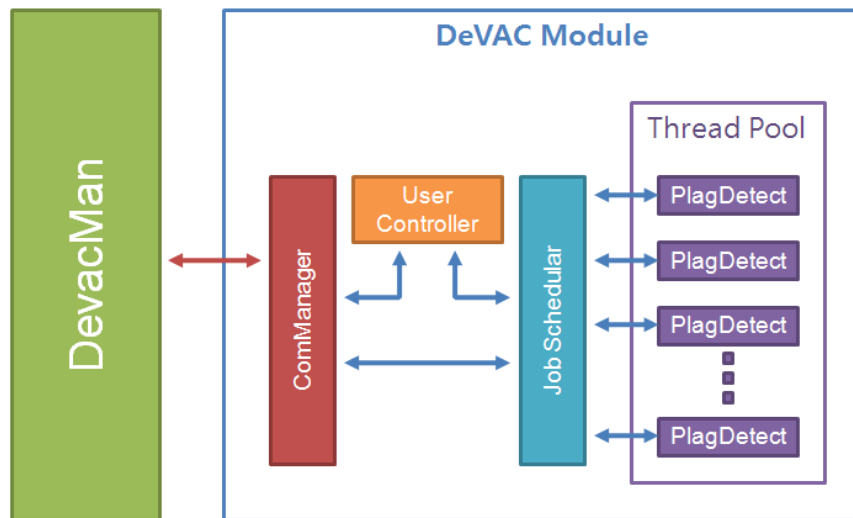


그림 2. 새로 개발하는 DeVAC 시스템. 필요 모듈을 분류하여 처리하며 Job Scheduler를 이용하여 표절탐색을 스케줄링하여 시스템의 안전성을 높인다.

그림 2는 새로 개발하는 DeVAC 시스템의 전체적인 구조를 나타내고 있다. ComManager를 이용하여 DevacMan과의 통신을 관할하며 User Controller는 사용자 정보 및 표절 탐색 권한을 확인하고 Job Scheduler를 이용하여 표절 탐색의 작업을 통제하여 시스템의 안정성을 높인다. 이는 많은 사용자가 이용할때 시스템의 안정성을 높여서 현재 DeVAC시스템의 잠재적인 위협을 제거할 수 있다.

3.1 DevacMan

DevacMan은 MFC 환경으로 DeVAC 시스템과 통신을 하는 모듈이다. 이를 통하여 사용자는 손쉽게 서비스를 이용할 수 있다. 웹 환경으로 충분히 지원할 수 있는 시스템이기는 하지만 전용 프로그램을 사용함으로써 사용자의 권한 및 버전 컨트롤을 쉽게 할 수 있다는 점이 강점이다. DevacMan에서 처리하는 일은 크게 다음과 같다.

- 사용자 로그인 및 로그오프. - bool Login(string userName, string passwd, userInfo& usrInfo)
- 사용자 정보 갱신. - bool UpdateUserInfo(userInfo usrInfo)

- 표절탐색 파일 생성. - bool makePlagFile(string fileName, string [] fileList)
- 표절탐색 파일 업로드 및 다운로드. - bool Uploadfile(string fileName, userInfo userInfo)
- 표절탐색 결과 확인. - bool checkPlagResult(string fileName)

DevacMan에서 현재 처리하고 있는 기능은 표절 탐색 파일 생성과 결과 확인이다. 즉, 사용자 정보와 관련된 작업과 업로드 및 다운로드는 웹서버에서 처리하고 있어 사용자에게 번거러움을 주고 있다. 새로 개선하는 DevacMan에서는 서버와의 연동을 통하여 사용자 정보 갱신 및 표절탐색 파일 업로드와 다운로드 및 현재 표절 탐색 진행상황을 사용자에게 알려주어 편의성을 도모한다.

3.2 ComManager

ComManager는 DevacMan과의 통신을 담당하는 모듈이다. 통신과 관련된 모듈을 따로 제공함으로써 시스템 보안과 네트워크 서비스를 안정적으로 제공하고 또한 시스템 변화가 있을때 빠르게 대처가 가능하다. ComManager에서 주로 제공하는 기능은 다음과 같다.

- DevacMan과의 통신.
- 보안 관련 서비스.
- 업로드된 표절탐색 파일 이동.

ComManager는 기본적으로 DevacMan과의 통신을 담당하지만 보안 관련 서비스도 제공하고 있다. 즉, 사용자가 불법적으로 접근을 하는 경우에 대비하여 서버와 연동하여 보안서비스를 제공한다. 또한 DevacMan에서 전송되어 온 표절 탐색 파일을 검사하고 실제 표절 탐색을 수행할 위치로 이동함으로써 표절 탐색을 좀더 용이하게 만들어준다.

3.3 User Controller

User Controller는 사용자 정보를 저장하고 각 모듈들에게 제공하는 역할을 한다. User는 기본적으로 웹서버를 통하여 회원 가입을 하게되고 회원정보를 수정할 수 있다. 이런 사용자 정보를 서버 모듈들에게 알려주는 역할을 하게되는데 사용자 정보는 표절 탐색 권한및 표절 탐색 위치를 결정하기 때문에 매우 중요한 정보이다. 이런 정보들을 다른 모듈들에게 제공하기 때문에 User Controller는 자주 사용이 되게 된다. User Controller에서 제공하는 기능은 다음과 같다.

- 사용자 정보 제공. - userInfo getUserInfo(string userName)
- 사용자 정보 갱신. - bool UpdateUserInfo(userInfo userInfo)
- 사용자별 표절 탐색 디렉토리 제공. - bool makePlagDir(string userName)

User Controller는 각 모듈들에게 사용자의 정보를 제공한다. 사용자 아이디와 패스워드를 이용하여 DevacMan에서 이루어지는 사용자 로гин을 제어하며 JobScheduler에 사용자의 권한을 알려주어 표절 탐색 스케줄링에 이용하도록 한다. 또한 사용자별 표절 탐색 디렉토리를 만들거나 제공하여 표절 탐색을 용이하게 도와준다.

3.4 Job Scheduler

Job Scheduler는 표절 탐색 작업을 스케줄링하고 표절 탐색을 수행하는 Plag Detect 모듈들을 관리한다. 표절 탐색 의뢰는 Job Scheduler에서 받아서 Plag Detect 모듈들에게 할당하는데 이는 Plag Detect 모듈들이 무분별하게 생성되어 시스템에 부하가 걸리는것을 막기 위해서이다. Job Scheduler에서 제공하는 기능은 다음과 같다.

- Plag Detect 모듈 관리.
- 표절탐색의뢰 관리.
- 표절탐색과 관련된 사용자 정보 갱신.

Plag Detect 모듈들은 Job Scheduler에서 Thread Pool로 관리가 된다. Thread Pool은 이미 생성된 Thread들을 유휴 상태로 만들어두고 서비스 요청이 있을때마다 유휴상태의 Thread들에 작업을 할당하여 처리하는 방식이다. 만일 유휴 상태의 Thread들이 존재하지 않으면 유휴 상태의 Thread가 생길때까지 작업은 대기상태로 있게된다. 이는 초기 부하는 기존의 시스템에 비해 크지만, 작업량이 많을 때에는 이미 생성된 Thread들만 사용하기 때문에 추가 부하는 크지 않다.

표절탐색의뢰 관리는 표절탐색 의뢰가 들어온 작업들에 대해 사용자 등급과 Thread Pool의 상태를 고려하여 작업을 할당하여 관리하는 기능이다. 이를 통하여 높은 사용자 등급의 경우 좀더 빠른 서비스를 제공받을 수 있는 기회를 가지게 된다. 표절탐색과 관련된 사용자 정보 갱신은 사용자가 표절 탐색을 위해 얼마나 시스템을 사용했는지 정보를 갱신하는 것으로 이를 통하여 사용자 등급에 따른 표절 탐색 서비스를 제공할 수 있게 된다.

3.5 Plag Detect

Plag Detect 모듈은 기존의 DEVAS 시스템의 라이트 버전이다. Plag Detect는 표절유무를 탐색하고 그 결과를 돌려주는 모듈이다. 하지만 DEVAS와는 다르게 N:N 표절 탐색이 아닌 1:1 탐색 및 1:N 탐색도 지원해서 좀 더 빠른 표절 탐색이 가능하다. Plag Detect 모듈을 따로 뚫으로서 표절 검사 방식이 개선되거나 테스트할 경우에 기존의 시스템을 수정하지 않아도 된다는 장점이 있다.

4 성능 검사

기존의 DeVAC시스템과 비교하여 새로 구성된 DeVAC시스템의 성능을 비교하여 성능의 변화를 체크할 필요가 있다. 기존의 시스템에 비해 성능이 크게 떨어지면 새로운 시스템을 사용하는것을 다시 고려하여야 하기 때문이다. 성능 검사는 대규모 표절 검사를 수행하였을때 걸리는 시간으로 측정하였다.

성능 변화를 체크하기 위해서 기존의 DeVAC시스템에서 어질수에 따른 시간 변화를 측정하고 동일 데이터를 새로운 DeVAC시스템을 이용하였을때를 비교 하였다.

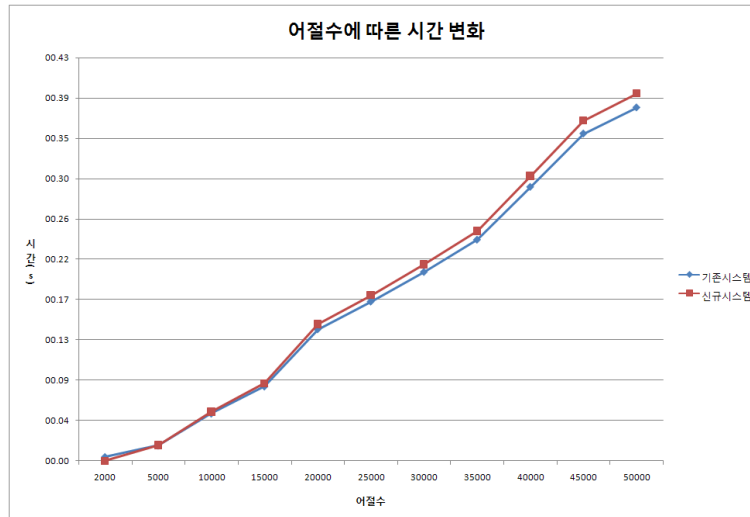


그림 3. 어절수에 따른 표절 검사 시스템의 시간 변화. 기존 시스템에 비해 미세하게 표절 검사에 걸리는 시간이 차이가 나는 것을 확인할 수 있다. 이는 기존 시스템이 성능에 주요 초점을 맞춘것에 비해 신규 시스템이 안정성과 차후 관리에 초점을 맞췄기 때문이다.

그림 3은 기존의 시스템과 신규시스템의 어절수에 따른 검사 시간을 비교한 그래프이다. 기존 시스템에 비해 신규 시스템은 미세하게 표절 검사에 좀더 시간이 더 걸리는 것을 확인할 수 있다. 이는 검사 어절수가 높아질수록 더욱 차이가 나는 것을 확인할 수 있는데 이는 기존의 시스템은 표절검사를 한개만 수행하여 시스템의 컴퓨팅 파워를 모두 사용한 경우이고, 신규 시스템의 경우 처음 시스템이 가동되면 Thread Pool 을 이용하여 컴퓨팅 파워를 분산 시켜 놓기 때문에 성능이 저하되는 것으로 볼 수 있다.

즉, 기존의 시스템에 비해 크기는 아니지만 성능이 조금은 낮아지는것을 확인할 수 있지만 이는 모듈화를 통한 업그레이드 및 유지보수의 용이성과 비교해 볼때 미비한 차이이며 오히려 성능이 개선되었다고 볼 수도 있다.

5 결론 및 향후 연구과제

본 보고서에서는 DeVAC 시스템의 모듈화를 통하여 차후 유지보수 및 업그레이드에 용이한 시스템을 제안하였다. 제안하고 실제로 구현한 방법은 DevacMan와 서버와의 좀더 긴밀한 연동을 통하여 사용자에게 좀더 편리하고 서버에는 좀더 안정적인 서비스를 제공하게 만들었으며 실제 성능도 크게 저하되지 않는 것을 확인하였다. 또한 각 기능들을 모듈화하여 제공함으로써 좀더 유연한 시스템을 구축하여 차후 개발에 용이하도록 하였다. 즉 본 보고서에서 제안하고 만들어진 시스템의 장점은 다음과 같다.

- DevacMan과 서버와의 좀더 긴밀한 연동을 통하여 사용자에게 좀더 편한 환경을 제공하였다.

- DeVAC 시스템의 모듈화를 통하여 차후 유지보수 및 업그레이드에 용이하도록 하였다.
- 서비스 제공 모델을 변경하여 DeVAC 시스템의 잠재적인 위협요소를 제거하였다.

본 보고서에서 제안한 시스템의 좀더 정확한 성능 측정을 위해서는 다음과 같은 실험이 이루어져야 한다. 기존의 실험은 서버에 단 하나의 표절 검사만 이루어짐을 가정하고 있다. 하지만 실제로 표절 검사는 여러명의 사용자가 동시에 수행하게 된다. 이는 현재 서버가 한 프로세스코어에 하나의 Devac 프로세서를 할당한다고 가정할때 현재 Devac 서버가 총 16 개의 프로세스코어를 가지고 있기 때문에 16명 이상 표절 검사를 수행하면 시스템의 성능이 극도로 나빠질 가능성이 있다. 즉, 동시 사용자가 증가함에 따른 표절 탐색 시간의 변화를 체크하여 기존의 시스템에 비해 얼마나 안정적인지를 측정해야 할 필요가 있다.

본 보고서에서 제안한 시스템의 성능을 개선하기 위한 향후 연구 과제로서, CUDA 를 이용하여 좀더 빠른 표절 탐색 서비스가 있다. 현재 CUDA 를 이용할 경우 표절 탐색의 핵심 기술인 Local Alignment 의 속도가 30 배 이상 좋아지는 것을 확인하였다. 이를 PlagDetect 모듈에 적용하여 DeVAC 시스템에서 좀더 빠른 서비스를 제공할 수 있을 것으로 기대된다. 하지만 CPU 프로세스 유닛의 수에 비해 CUDA 가 사용하는 GPU 는 좀더 작고 Thread 를 이용하여 Job Scheduling 을 수행하면 성능이 급격하게 떨어지는 것으로 알려져 있으므로 GPU 로 처리할 작업을 분류하는 작업이 선행되어야 한다. 즉 Job Scheduler 모듈에서 CUDA 를 이용하여 표절 검사를 수행할 작업들을 분류하고 CUDA 를 이용한 PlagDetect 모듈들에게 할당하는 방법과 CUDA 를 이용한 PlagDetect 모듈의 개발이 필요하다.

참고 문헌

1. 류창건, "Affine gap penalty model 을 이용한 한글 표절 탐색 시스템(devac) 표절 탐색 성능 향상," Technical Report SYS08002, Graphics Application Lab. CS Dept., January 2008.
2. 김형준, "Devac 서버의 신규 구축 및 성능 비교," Technical Report SYS08004, Graphics Application Lab. CS Dept., March 2008.
3. 류창건, "Xtreme toolkit pro 9.7을 이용한 윈도우용 sovac 클라이언트 프로그램 제작하기," Technical Report PRJ08002, Graphics Application Lab. CS Dept., May 2008.