실시간 모니터링 시스템을 이용한 그리드 딜리버리 서비스 검출 방법

김연어*, 우균*
*부산대학교 컴퓨터공학과
e-mail:{yeoneo, woogyun}@pusan.ac.kr

A Detecting Method for Grid Delivery Services Using Real-Time Monitoring System

Yeoneo Kim*, Gyun Woo*
*Department of CSE, Pusan National University

요 약

최근 소프트웨어는 네트워크를 이용한 서비스가 대부분이다. 그 중 그리드 딜리버리는 서버/클라이언트 모델에서 서버의 부담을 줄이기 위해 개발된 서비스다. 그리드 딜리버리는 사용자의 자원을 이용해 서비스를 원활하게 제공하는 방법으로 충분히 악용할 수 있다. 그러므로 이 논문은 악용되고 있는 그리드 딜리버리를 탐색하는 방법을 제안한다. 제안하는 방법은 모니터링 도구를 통해 모든 프로세스의 네트워크 자원을 감시한다. 그리고 네트워크가 사용되는 프로세스를 그리드 딜리버리로 추정하는 방법이다. 제안 방법은 두 단계로 나뉘어 진행된다. 처음으로 네트워크 자원을 사용하는 프로세스를 후보로 뽑아낸다. 두 번째로 이전 단계의 후보에 필터링을 적용해 그리드 딜리버리로 의심되는 프로세스를 찾아낸다. 이 논문에서는 제안하는 방법으로 실제 그리드 딜리버리 프로세스가 검출되는 것을 보임으로 제안하는 방법이 유효함을 보인다.

1. 서론

최근 소프트웨어는 네트워크를 통한 서비스가 대부분이다. 이러한 서비스는 주로 서버/클라이언트 모델을 기반으로 하여 제공되고 있다. 서버/클라이언트 모델은 사용자가 많을수록 서버에 부담이 증가하는 모델로서, 서버의 유지비용이 많이 든다. 이러한 문제를 해결하기 위해 나온 방법의 하나로 그리드 컴퓨팅이 있다.

그리드 컴퓨팅은 분산 컴퓨팅(distributed computing)의 일종이다[1]. 그리드 컴퓨팅은 분산된 컴퓨터들의 자원을 사용해 계산 능력이 높은 하나의 거대한 가상 머신과 같이 사용할 수 있게 하는 기법이다. 그리드 컴퓨팅은 현재 다양한 소프트웨어 서비스에서 이용되고 있으며 특히 그리드 딜리버리(grid delivery)라는 서비스를 통해 여러 분야에서 이용되고 있다. 하지만 그리드 딜리버리는 악용할 소지가 많은 서비스이다.

그리드 딜리버리는 사용자들의 자원을 이용해서 서비스를 원활하게 제공하는 방법이다[2,3]. 이 방법은 소프트웨어 서비스를 위한 프로그램 설치 시 같이 설치되어 사용자의 자원을 이용하여 서비스를 원활하게 제공한다. 하지만 이 방법은 서비스가 종료된 이후에도 사용자 몰래 컴퓨터의 자원을 훔쳐 서비스를 제공할 수 있는 문제점이 있다. 대부분의 그리드 딜리버리 프로그램은 사용 약관에 적혀있는데 약관을 제대로 읽지 않는 사용자는 어떤 서비스인지 모르고 설치할 수 있다.

본 연구는 지식경제부 산업원천기술개발사업의 일환으로 수행하였음. [10035185, 서버 기반 SW 서비스의 분할 실행 기술 개발]

이 논문에서는 이처럼 사용자 몰래 설치되거나 악용되는 그리드 딜리버리 서비스를 검출하는 방법을 제안한다. 기존의 그리드 딜리버리 서비스를 검출하는 방법은 DB에 기반을 둔 검출 방법이기 때문에 변종이나 새로운 그리드 딜리버리 서비스는 검출하지 못하는 단점이 있다. 이 논문에서는 이러한 단점을 해결하기 위해 실시간 감시 시스템을 이용하는 검출 방법을 제안한다.

이 논문의 구성은 다음과 같다. 2장에서는 그리드 딜리버리 서비스에 대해 살펴보고 기존의 해결방법을 살펴본다. 그리고 3장에서는 모니터링 도구를 이용해 그리드 딜리버리 서비스를 검출하는 방법에 대해 알아본다. 그리고 4장에서 결론을 맺는다.

2. 연구 배경

2.1. 그리드 딜리버리 서비스

그리드 딜리버리 서비스는 그리드 컴퓨팅의 개념에서 출발한 방법으로 사용자들의 계산 자원을 활용한다[2,3]. 이 방법은 기존의 서버/클라이언트 모델의 단점인 트래픽 집중이나, 병목 현상을 피할 수 있으며 대용량 소프트웨어 서비스에 적합한 모델이다. 또한, 이 방법은 서비스의 품질(QoS : Quality of Service)을 보장하는데 도움을 준다.

그래서 최근 많은 업체에서 그리드 딜리버리 서비스를 이용해 다양한 서비스를 제공하고 있다.

그리드 딜리버리는 크게 두 종류의 업체에서 많이 이용되고 있다. 첫 번째는 인터넷 웹 하드 업체이다. 웹 하드 업체의 특성상 대용량 데이터 전송이 빈번히 발생하기 때문에 서버/클라이언트 모델보다 그리드 딜리버리 서비스를 이용하는 것이 효과적이다. 두 번째는 실시간 스트리밍 서비스를 제공하는 업체이다. 실시간 스트리밍을 제공하는 업체는 데이터를 계속 사용자들에게 보내기 때문에 같은 데이터를 이용하는 고객끼리 자원을 공유하면 서버에 부담이 줄고 원활한 서비스 제공을 할 수 있어 많이 사용된다.

그리드 딜리버리는 QoS 측면에서 효과적이지만 문제점도 존재한다. 그리드 딜리버리 서비스의 근본적인 문제는 사용자의 계산 자원을 이용하기 때문에 사용자는 이를 정확히 알고 있어야 한다는 점이다. 현재 많은 그리드 딜리버리를 사용하는 업체는 사용자의 자원을 사용하는 사실을 이용약관에 포함하고 있다. 하지만 대부분 사용자는 프로그램 설치 시 이용약관을 잘 읽지 않아 이를 알지 못하는 경우가 많기에 유명무실한 상황이다. 또한, 그리드 딜리버리 서비스를 악용하는 것도 문제가 된다. 원래 그리드 딜리버리는 사용자가 서비스를 종료하면 사용자의 계산 자원 사용을 중지하여야 하지만 이를 지키지 않고 계속 프로세스에 상주하여 계산 자원을 훔쳐 사용하는 경우 문제가 된다.

이러한 그리드 딜리버리 서비스의 문제는 주로 웹 하드 업체를 통해 많이 발생하고 있다[4]. 최근 웹 하드 업체는 식당이나 편의점과 같이 많은 사람이 이용하는 공간에 무료 쿠폰을 배부하고 있으며 사용자는 무료 쿠폰이라는 말에 웹 하드 업체의 프로그램을 이용하는 경우가 늘고 있다. 이러한 웹 하드 업체는 대부분 그리드 딜리버리 서비스를 이용하여 사용자의 자원을 이용하여 서비스를 제공한다. 하지만 많은 웹 하드 업체는 그리드 딜리버리를 악용하여 사용자가 프로그램을 종료하였음에도 계속 사용자의 자원을 훔쳐 사용하는 경우가 많다. 이러한 사실을 모르는 사용자는 원인 모르게 컴퓨터가 느려졌다고 생각하게 되는 문제가 발생한다.

2.2. 기존의 해결 방안

그리드 딜리버리는 바이러스로 분류되지 않기 때문에 기존의 백신 프로그램으로는 판별할 수 없다. 그래서 그리드 딜리버리를 검출하고 치료하는 별도의 프로그램이 개발되었다[5,6]. 이러한 기존의 검출 방법은 접근 방안에 따라 두 가지로 나누어 볼 수 있다.

첫 번째는 그리드 딜리버리 서비스의 이름을 DB로 저장하고 프로세스의 이름을 비교하는 것으로 검출하는 방법이다. 이 방법은 잘 알려진 그리드 딜리버리 서비스를 정확하게 제거할 수 있는 장점이 있다. 하지만 새로운 그리드 딜리버리나 변종 그리드 딜리버리가 나오게 되면 검출하지 못한다. 또한, 단순 프로세스 이름을 비교하는 방법이기 때문에 일반 프로세스를 그리드 딜리버리로 판단할 수 있다. 이와 같이 DB를 이용해서 검출하는 대표적인 프로그램으로 Gridswitch가 있다[5].

두 번째는 윈도우의 기본 프로세스 이외의 모든 프로세스를 차단하는 방법이다. 이 방법은 그리드 딜리버리 프로세스뿐만 아니라 알려지지 않거나 안전하지 않은 모든 프로세스를 차단할 수 있는 장점이 있다. 하지만 이 방법은 윈도우 기본 프로세스 이외의 안전한 프로세스까지 모두 제거해서 원치 않는 프로세스까지 같이 종료된다. 대표적인 프로그램으로 프로세스 클린이 있다[6].

3. 실시간 모니터링 시스템

이 논문에서는 그리드 딜리버리 서비스를 검출하기 위해 실시간 모니터링 도구를 이용하는 방법을 제안한다. 실시간 모니터링 도구는 프로그램의 실행 상태를 관찰할 수 있게 하는 프로그램으로 다양한 컴퓨터 자원을 실시간으로 보여주는 도구이다. 제안하는 방법은 그리드 딜리버리 서비스의 특징인 사용자의 자원을 이용하는 점에 착안해 그리드 딜리버리 프로세스를 검출하고자 한다.

이 논문에서는 다양한 분야에서 사용되는 그리드 딜리버리 서비스 중 웹 하드와 같은 파일을 공유하고 사용하는 그리드 딜리버리 서비스를 검출한다. 대부분의 웹 하드 업체에서는 서버의 비용 때문에 그리드 딜리버리 서비스를 이용한다. 하지만 웹 하드는 다른 분야의 그리드 딜리버리와 달리 사용자가 프로그램을 종료한 이후에도 컴퓨터에 상주하며 다운받았던 파일을 다른 사용자와 공유하는 사례가 많기 때문이다.

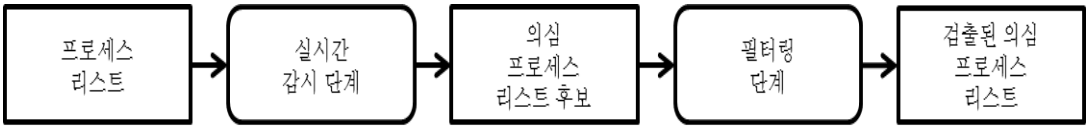
웹 하드와 같은 그리드 딜리버리 서비스는 사용자의 자원 중 CPU, 디스크, 네트워크 자원 등을 이용한다. 이 중 CPU 자원은 비교적 많이 사용되지 않으며 대개 파일을 가져가기 위해 디스크 자원과 파일 전송을 위해 네트워크 자원을 이용한다. 하지만 디스크 읽기로 읽힌 파일은 네트워크를 통해 다른 사용자에게 전송된다. 그러므로 이 논문에서는 프로세스의 네트워크 사용량만을 고려하여 그리드 딜리버리 프로세스를 검출하는 방법을 제안한다.

3.1. 전체 분석 흐름

그림 1은 제안 방법의 전체적인 분석 흐름을 나타낸다. 그림 1과 같이 제안하는 시스템은 입력으로 현재 실행되고 있는 프로세스 리스트를 받는다. 그리고 실시간 감시 단계에서는 모니터링 도구를 이용해서 그리드 딜리버리로 의심되는 프로세스의 후보를 선정한다. 이후 필터링 단계에서는 의심되는 프로세스 후보에서 안전하다고 여겨지는 프로세스들을 제거하는 단계로 구성된다.

3.2. 실시간 감시 단계

실시간 감시 단계에서는 실행되고 있는 프로세스를 대상으로 그리드 딜리버리 서비스로 의심되는 프로세스 후



(그림 2) 그리드 딜리버리 서비스 검출을 위한 전체적인 흐름도. 제안 방법은 프로세스 리스트를 입력으로 받아 실시간 감시 단계에서 모니터링 도구를 이용해 의심 프로세스 리스트 후보를 뽑아낸다. 그리고 필터링을 통해 의심 프로세스 리스트를 생성한다.

보를 선정한다. 이 단계에서는 정해진 시간 동안 전체 프로세스를 대상으로 네트워크 사용량을 수집한다. 그리고 네트워크 사용량이 조금이라도 있는 프로세스를 후보로 선정한다. 이 단계에서는 실시간으로 프로세스의 네트워크 사용량을 감시하기 위해 모니터링 도구인 Process Hacker를 변형해서 정보를 수집한다.

Process Hacker는 오픈소스로 제공되는 모니터링 도구로서 다양한 자원을 관측할 수 있다[7]. 또한, Process Hacker는 특정 시간의 전체 프로세스의 자원 정보를 CSV 파일로 저장하는 기능을 제공한다. 하지만 현재 Process Hacker에서 실시간으로 프로세스의 자원 정보를 수집하여 저장하는 것은 불가능하다. 그래서 이 논문에서는 Process Hacker를 수정하여 일정 시간 동안 네트워크 사용량을 수집하여 저장할 수 있도록 변경하였다. 그리고 저장된 데이터에서 네트워크 사용량이 있는 프로세스들만 그리드 딜리버리 프로세스로 의심되는 후보로 선정한다. 그 결과로 얻어낸 프로세스 후보가 표 1과 같다.

<표 1> 그리드 딜리버리 프로세스로 의심되는 프로세스 후보 리스트. 모든 프로세스를 대상으로 10분간 감시하였을 때 네트워크를 사용한 프로세스의 리스트이다.

의심 프로세스 후보
chrome.exe
Dropbox.exe
firefox.exe
iexplore.exe
natsvc.exe
SuperDownService.exe
svchost.exe
System

표 1은 전체 모니터링 도구를 이용해 얻어낸 그리드 딜리버리 서비스로 의심되는 후보이다. 표 1은 10분간 모든 프로세스를 감시하여 그 중 네트워크 사용량이 있는 프로세스를 선택한 결과이다. 이후 표 1의 결과는 필터링 단계로 전달되어 정확한 결과를 얻는다.

3.3. 필터링

필터링 단계는 이전 단계에서 얻어진 결과에 오경보(false alarm)를 줄인다. 이전 단계의 결과는 네트워크를 사용량이 있는 모든 프로세스를 그리드 딜리버리 프로세스를 후보로 선정하였다. 하지만 그리드 딜리버리 프로세스 이외에도 네트워크를 사용하는 프로세스는 존하기 때문에 이러한 프로세스는 후보에서 제거해줘야 한다. 이 단계에서는 그리드 딜리버리가 아니지만, 네트워크를 사용하는 프로세스를 크게 3종류로 나누어 필터링에 등록한다.

첫 번째는 윈도우 자체가 사용하는 프로세스이다. 윈도우는 자동 업데이트나 다른 기능 때문에 자체적으로 네트워크를 사용하는 프로세스를 가지고 있다. 이러한 프로세스는 그리드 딜리버리 프로세스가 아니므로 필터링 과정에서 제거해줘야 한다.

두 번째는 일반 프로세스이다. 이 프로세스는 윈도우 기본 프로세스에 속하지 않지만, 일반적으로 널리 사용되는 프로세스인 웹 브라우저와 같은 프로그램이 포함된다. 이 종류는 네트워크를 사용하면서 많이 사용되는 일반 프로세스가 포함된다. 제안하는 방법에서는 주로 웹 브라우저가 이 종류에 포함된다.

세 번째는 사용자 정의 프로세스이다. 이 필터는 위의 두 종류의 필터에 속하지 않지만 사용자 필요로 직접 설치된 프로세스들이 여기에 속한다. 두 번째 필터인 일반 프로세스와 다른 점은 HTTP나 PoP3, IMAP, FTP를 사용하는 프로세스가 아닌 프로세스이다. 세 번째는 필터는 다른 필터와 달리 사용자가 직접 추가할 수 있다. 위의 세 필터를 적용한 결과는 표 2와 같다.

<표 2> 프로세스 필터링 리스트. 이미 안전하다고 판단된 프로세스 리스트로서 윈도우 기본, 웹 브라우저, 사용자 정의 프로세스로 구성된다.

필터링 종류	프로세스 명
윈도우 기본 프로세스	svchost.exe
	System
	wuauclt.exe
일반 프로세스	chrome.exe
	firefox.exe
	iexplore.exe
사용자 정의 프로세스	Dropbox.exe

표 2는 필터링 단계에서 걸러지는 프로세스 리스트의 예이다. 표 2에서 윈도우 기본 프로세스로 걸러지는 프로세스는 System과 svchost, wuauclt 가 있다. System은 시스템 커널 모드 스레드가 동작하는 프로세스이고, svchost는 DLL을 하나의 서비스로 제공하는 윈도우 기본 프로그램이며, wuauclt는 윈도우 자동 업데이트를 위한

프로세스이다. 그리고 일반 프로세스의 프로세스는 인터넷 브라우저로써 chrome는 구글의 크롬이며, iexplore는 MS의 인터넷 익스플로러이며, firefox는 모질라의 파이어 폭스이다. 마지막으로 사용자 정의 프로세스에는 웹 하드 프로그램인 드랍박스의 프로세스 Dropbox를 추가하여 필터링 되도록 하였다.

3.4. 실험결과

앞선 두 단계를 거쳐 나온 그리드 딜리버리 프로세스로 의심되는 프로세스는 표 3과 같다. 표 3의 결과는 표 1의 의심 프로세스 후보에서 표 2의 필터링을 적용한 결과이다. 실험 환경에서 natsvc.exe 프로세스는 웹 하드 업체인 파일시티와 본디스크에서 사용하는 그리드 딜리버리 프로세스이다. 그리고 SuperDownService.exe 프로세스는 웹 하드 업체인 슈퍼다운에서 사용하는 프로세스이다. 실험 환경에서 설치된 웹 하드 프로그램은 파일시티, 본디스크, 슈퍼다운, 드랍박스와 같은 4종류가 설치되었다. 하지만 드랍박스는 그리드 딜리버리 프로세스를 사용하지 않는 프로그램이기 때문에 필터링 단계에서 걸러졌기 때문에 표 3과 같이 2종류의 프로세스가 검출되었다.

<표 3> 그리드 딜리버리 프로세스로 검출된 프로세스. 전체 51개의 프로세스 중 2개의 프로세스가 그리드 딜리버리 프로세스로 검출되었으며 각 프로세스는 파일시티와 본디스크, 슈퍼다운 웹 하드 업체와 연관 있는 것으로 나타났다.

의심 프로세스	연관된 웹 하드 업체
natsvc.exe	파일시티, 본디스크
SuperDownService.exe	슈퍼다운

4. 결론

이 논문에서는 그리드 딜리버리 서비스의 문제점에 대해 알아보고 그리드 딜리버리 서비스를 탐색할 수 있는 새로운 방법을 제안하였다. 제안하는 방법은 두 단계에 걸쳐 실행되는 방법으로 첫 번째로 프로세스의 자원 중 네트워크 자원을 사용하는 프로세스를 그리드 딜리버리 프로세스 후보로 선정한다. 그리고 두 번째로 필터링을 통해 안전한 프로세스를 걸러낸다. 이후 나온 결과가 그리드 딜리버리 프로세스로 검출된 프로세스이다. 이 논문에서는 전체 51개의 프로세스를 대상으로 제안하는 방법을 적용하였다. 그리고 그 결과로 2개의 그리드 딜리버리 서비스를 검출함으로 제안하는 방법이 타당함을 보였다.

향후 연구 과제로서 중요한 것은 제안하는 방법의 정확도를 높이는 연구이다. 현재는 필터링의 정확도와 양이 오경보의 수준을 결정한다. 그러므로 필터링 이외에도 정확도를 높이는 방법에 관한 연구가 필요하다. 그리고 제안하는 방법은 사용자의 암호를 훔치는 키로그 같은 프로그램 검출에도 유용할 것으로 예상된다. 그러므로 향후 키로그 프로그램을 검출하는 방법에 적용해 볼 예정이다.

참고문헌

[1] Ferreira, Luis, et al. “Introduction to grid computing with globus,” IBM Corporation, International Technical Support Organization, 2003.

[2] Lv, ZhiHui, ShiYong Zhang, and YiPing Zhong. “Research on service model of content delivery grid,” Advanced Web Technologies and Applications (2004): 321-330.

[3] 문의선, “콘텐츠 전송의 혁신 ‘그리드 딜리버리’”, 데이터넷, http://www.datanet.co.kr/, 2008.

[4] 민상식, “P2P 무료영화 함부로 다운받지 마세요”, 해럴드경제, http://nbiz.heraldcorp.com/, 2012.

[5] Gridswitch, “Simple is Smart Gridswitch”, http://www.gridswitch.co.kr/, (2013년 03월 15일 방문).

[6] 정대원, “프로세스크랙”, http://www.processcrack.co.kr, (2013년 03월 15일 방문).

[7] Wen Jia Liu, “Process Hacker,” http://processhacker.sourceforge.net/, (2013년 03월 15일 방문).