

스토리지 장치의 시계열 데이터 및 델타 측정을 이용한 시스템 에이징 원인 분석 및 측정 도구 개발

System Aging Cause Analysis and Measurement Tool Development Using Time Series Data and Delta Measurement of Storage Devices

장은태 (Eun-tae Jang) (제1저자) | 세종대학교 정보보호학과 석사과정 | euntaejang@gmail.com
박기웅 (Ki-Woong Park) (교신저자) | 세종대학교 정보보호학과 부교수 | woongbak@sejong.ac.kr

목 차	1. 서론	4. 스토리지 데이터 기반의 에이징 측정도구를 통한 실험
	2. 관련 연구	4.1 실험환경 구축
	3. 스토리지 장치의 시계열 데이터 및 델타 측정을 위한 프레임워크 디자인	4.2 실험 방법
		4.3 실험 결과
		5. 결론

초 록

컴퓨터 시스템을 장기간 사용하게 되면 시간이 지남에 따라 성능이 저하되고 오류 발생률이 증가하는 경향이 있는데, 이러한 현상을 일반적으로 '시스템 에이징'이라고 한다. 컴퓨터 시스템 에이징 현상의 원인을 규명하기 위하여 운영체제의 커널 영역에서부터 사용자 애플리케이션 영역에 이르기까지의 다양한 관련 연구들이 수행되었으며 시스템 성능 저하의 정확한 원인을 찾는 것은 매우 어려운 연구로 여겨지고 있다. 본 논문에서는 컴퓨터 시스템 재 반복된 소프트웨어 설치/제거에 따른 컴퓨터 시스템의 성능저하에 대한 가설을 수립하고 이를 정량적으로 측정하기 위해 설치된 소프트웨어가 제거될 경우 메모리, 레지스트리, 디스크 등 여러 요소들이 소프트웨어 설치 전의 운영체제와 같이 복구 되지 않는 것을 확인하였으며 이를 통해 소프트웨어 설치 및 제거 작업과 시스템 에이징의 연관성을 연구하였다. 검증된 가설의 내용을 바탕으로 컴퓨터 시스템의 스토리지 데이터에 기반한 시스템 에이징 측정 도구를 개발하였으며, 개발 된 측정 도구를 사용하여 소프트웨어 설치 전, 설치 후, 제거 후의 디스크를 덤프하고 덤프파일에서 파일들의 리스트를 추출한 뒤 소프트웨어가 제거될 때 디스크에 남겨진 터미파일들의 개수 및 크기에 대한 정량적 결과를 도출하였으며 이러한 제거 되지 않은 터미 파일이 시스템에 영향을 미칠 수 있음을 실험을 통해 확인하였다. 더 나아가 논문에서 개발된 도구가 제거된 소프트웨어에 대한 터미파일을 식별하고 스토리지에 남겨진 터미파일들을 통해 시스템 에이징 측정에 활용될 수 있음을 제시한다.

* 키워드 : 시스템 에이징, 성능, 스토리지, 디스크, 소프트웨어 설치 및 제거, 측정 도구

ABSTRACT

Long-term use of computer systems tends to reduce performance over time and increase error rates, which is commonly referred to as system aging. In order to identify the cause of computer system aging, various related studies have been conducted from the kernel area of the operating system to the user application area, and it is considered to be a very difficult study to find the exact cause of system performance degradation. In this paper, we set up a hypothesis about the performance degradation of computer system according to repeated software installation/removal of computer system and confirmed that various elements such as memory, registry, disk are not restored like operating system before software installation when installed software is removed to measure it quantitatively. Based on the verified hypothesis, a system aging measurement tool based on storage data of a computer system was developed. The developed measurement tool was used to dump the disk before, after installation, after removal, and extract the list of files from the dump file. When the software was removed, quantitative results were obtained on the number and size of pile files left on the disk. Furthermore, it is suggested that the developed tools can be used for system aging measurement through dummy files left in storage and identifying dummy files for the removed software.

* Keywords : system aging, performance, storage, disk, software installation and removal, measurement tool

* 본 논문은 2020년도 과학기술정보통신부의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2018-0-00420, API 호출 단위 자원 할당 및 사용량 계량이 가능한 서버리스 클라우드 컴퓨팅 기술 개발)

• 논문접수일 : 2020년 8월 7일 • 최종심사일 : 2020년 8월 8일 • 게재확정일 : 2020년 8월 28일

1. 서론

시스템을 장기간 사용하게 되면 시간이 지남에 따라 성능이 저하되고 오류 발생률이 증가하는 경향이 있는데, 이러한 현상을 일반적으로 시스템 에이징이라고 한다. (Kourai, Kenichi, and Shigeru Chiba. "A fast rejuvenation technique for server consolidation with virtual machines."). 운영체제에서의 에이징 원인을 분석하기 위하여 운영체제의 커널 영역에서부터 사용자 애플리케이션 영역에 이르기까지의 여러 관련 연구들이 존재하며 정확한 시스템 성능 저하의 원인을 찾는 것은 매우 어려운 연구로 여겨지고 있다(Cotroneo, Domenico, et al. "Software aging analysis of the linux operating system."). 본 논문에서는 컴퓨터 시스템 재 반복된 소프트웨어 설치/제거에 따른 컴퓨터 시스템의 성능 저하에 대한 가설을 수립하고 이를 정량적으로 측정하기 위해 설치된 소프트웨어가 제거될 경우 메모리, 레지스트리, 디스크 등 여러 요소들이 소프트웨어 설치 전의 운영체제와 같이 복구되지 않는 것을 확인하였으며 이를 통해 소프트웨어 설치 및 제거 작업과 시스템 에이징의 연관성을 연구한다. 검증된 가설을 바탕으로 설치된 소프트웨어가 제거될 경우 임시파일, 로그 파일, 링크 파일 등의 많은 파일들을 디스크에 남기고 제거되는 것을 확인하며 시스템 에이징 현상과 소프트웨어 설치 및 제거와의 연관성을 연구하기 위해 시스템의 스토리지 데이터에 기반한 시스템 에이징 측정 도구를 개발한다. 개발된 도구를 사용하여 소프트웨어 설치 전, 설치 후, 제거 후의 디스크를 덤프하고 덤프 파일에서 디스크에 저장된 파일들을 추출한 뒤 소프트웨어가 제거될 때 제거되지 않고 디스크에 남겨진 더미 파일의 개수와 크기에 대한 정량적 결과를 도출하며, 이러한 제거되지 않은 더미 파일들이 시스템의 성능에 영향을 미칠 수 있음을 실험을 통해 확인한다. 그리고 더 나아가서 논문에서 개발된 도구가 제거된 소프트웨어에 대한 더미 파일을 식별하고 스토리지에 남겨진 더미 파일들을 통해 시스템 에이징 측정에 활용될 수 있음을 제시한다. 본 논문의 구성은 다음과 같다. 2장에서는 시스템을 대상으로 에이징 현상의 원인을 연구한 기존 관련 연구를 정리하고, 3장에서는 시스템 에이징 측정 도구 개발을 위한 시스템 디자인을 설명하며, 4장에서는 본 논문에서 개발한 시스템 에이징 측정 도구를 통한 실험 방법과 실험 결과를 도출한다. 5장에서는 결론 및 추후 연구 방향을 제시한다.

2. 관련 연구

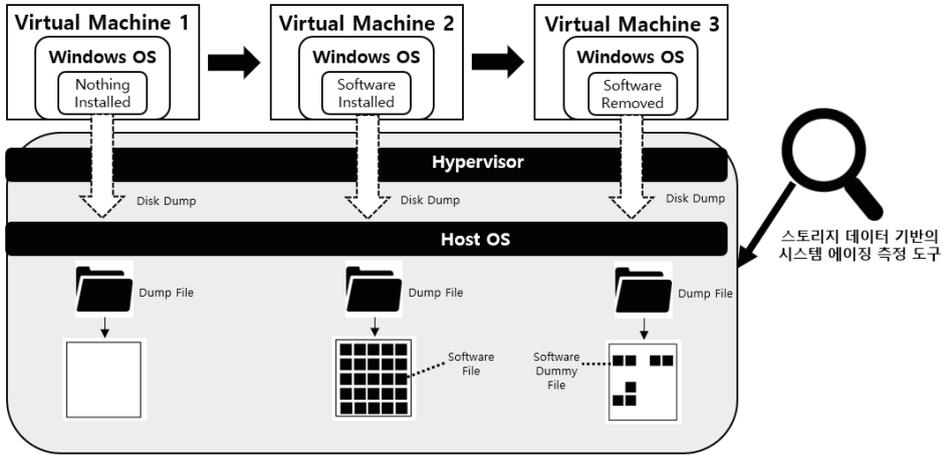
본 장에서는 시스템에서 발생하는 에이징 현상의 원인을 찾기 위하여 수행한 기존 연구들에 대해 설명하고 그에 관한 한계점을 도출한다. 연구자들은 에이징 현상의 원인을 연구하기 위하여 소프트웨어부터 커널에 이르기까지 다양한 대상에 대하여 연구를 수행하였다. "Software Aging Analysis of the Linux Operating System"의 저자 Domenico Cotroneo은 리눅스 시스템에서 오랜 시간 동

안 지속적으로 실행되는 소프트웨어가 시간에 지남에 따라 성능이 저하되며 오류 발생률이 증가하는 것을 확인하였으며 이에 대한 원인을 분석하기 위해 커널 추적 도구를 개발하여, 여러 커널 하위 시스템의 관련 매개 변수를 수집하고 통계 분석을 통해 리눅스 운영체제의 에이징 원인을 확인하였다. 실험 결과 원인은 Process Management, File system이 핵심 원인을 도출하였다(Cotroneo, Domenico, et al. "Software aging analysis of the linux operating system."). 또한 "Analysis of Software Aging in a Web Server"의 저자 Michael Grottk은 장시간 실행되는 소프트웨어 시스템의 고장률이 증가하거나 성능이 점진적으로 저하되는 현상이 많이 목격되며, 이러한 현상은 내부 오류의 누적 및 운영체제의 리소스 고갈이 원인이 되는 소프트웨어 에이징 현상이라고 하였으며(Grottk, Michael, et al. "Analysis of software aging in a web server."), "A Methodology for Detection and Estimation of Software Aging"의 저자 Sachin Garg과 Aad van Moorsel은 소프트웨어를 장기간 동안 사용하면서 소프트웨어의 잦은 오류를 경험하였으며 운영체제의 메모리 스왑 공간 소진, 운영체제의 리소스 사용량 증가가 소프트웨어 에이징의 핵심적인 원인이 된다고 하였다(Garg, Sachin, et al. "A methodology for detection and estimation of software aging."). 기존 시스템 에이징 관련 연구들은 소프트웨어를 오랫동안 사용하였을 때의 성능 저하에 대하여 연구를 수행하였으며 시스템에서 발생하는 에이징 현상의 원인을 소프트웨어를 장기간 사용하게 되었을 때의 성능저하에 초점을 맞추었다. 따라서 해당 실험을 위해서는 실험환경 구축을 위해 장시간 소프트웨어를 실행 시켜두어야 했다. 본 논문에서는 시스템을 장기간 사용하게 되었을 때의 성능저하에 대한 원인이 소프트웨어의 설치 및 제거와 밀접한 관련이 있음을 제시하며 스토리지 데이터에 기반한 시스템 에이징 측정 도구를 개발하여 이를 확인한다. 본 논문에서 제안하는 도구는 소프트웨어나 시스템의 장기간의 관찰 없이 소프트웨어 설치/제거에 의해 생기는 더미 파일을 통해 시스템 에이징을 측정함으로써 효율적이다.

3. 스토리지 장치의 시계열 데이터 및 델타 측정위한 프레임워크 디자인

본 장에서는 운영체제에서 소프트웨어가 제거될 경우, 소프트웨어가 완벽하게 제거되지 않고 많은 더미 파일들을 하드디스크에 남기고 제거되는 것을 확인하고, 남겨진 파일들이 시스템의 성능에 영향을 미치는 정도를 측정하기 위한 에이징 측정 도구를 디자인한다. 해당 도구는 파이썬 언어를 사용하여 개발되며 소프트웨어 제거 후 남겨진 더미 파일들을 확인하기 위해 디스크를 덤프할 수 있으며 덤프 파일에서 운영체제 내의 파일을 추출한다. 소프트웨어가 설치되어 있을 상태와 제거된 상태의 더미 파일을 추출하고 이를 비교하여 제거되지 않은 파일들의 개수와 크기를 추출한다.

해당 도구의 기능은 사용자가 원하는 시점에서 디스크 덤프가 가능하며, 디스크 덤프 파일에서 파



<그림 1> 스토리지 데이터 기반의 시스템 에이징 측정도구를 통한 데이터 측정 및 추출

일 리스트에 대한 정보를 추출할 수 있다. 또한 서로 다른 두 개이상의 디스크 덤프 파일에서 추출한 파일들의 리스트를 비교하여 차이를 도출할 수 있으며, 소프트웨어 설치 전, 설치 후, 제거 후로 디스크 상태를 나눈 뒤 도구를 사용하여 각각의 디스크를 덤프하고 세 가지 덤프 파일에 대한 파일 리스트들을 추출한 뒤 이를 비교하여 소프트웨어 제거 후에 디스크에 남아있는 파일들에 대한 개수 및 크기의 정량적 결과를 도출한다. 도출된 결과를 바탕으로 제거되지 않은 더미 파일이 시스템에 영향을 미치는 정도를 측정한다.

4. 스토리지 데이터 기반의 에이징 측정 도구를 통한 실험

본 장에서는 개발된 시스템 에이징 측정 도구를 바탕으로 실험을 수행한다. 개발된 도구는 소프트웨어 제거 후 남겨진 더미 파일들을 확인하기 위해 디스크 덤프, 운영체제 내의 파일 추출, 추출된 파일의 비교를 수행하여 소프트웨어가 설치되어있을 상태와 제거된 상태를 비교한다. 실험 대상 소프트웨어는 <그림 2>과 같다.

4.1 실험 환경 구축

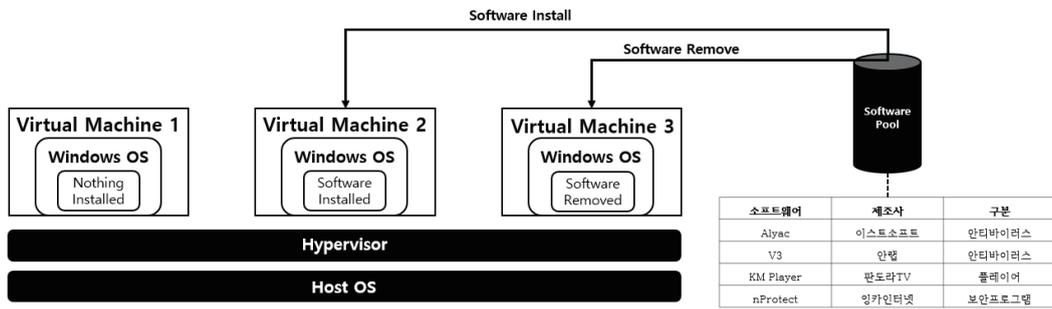
개발된 시스템 에이징 측정도구에 대한 실험에서 디스크 덤프 및 추출한 파일들의 정보를 분석하기 위해 가상환경(VM)을 사용하여 환경을 구축하였다. 가상환경을 사용하여 환경을 구축한 이유는 가상 머신에서 지원해주는 스냅샷 등 여러 유용한 기능들을 활용하여 효율적으로 소프트웨어 설치 전, 설치 후, 제거한 뒤의 상태의 운영체제 내에 존재하는 파일들을 분석할 수 있기 때문이다. 실험을 위해

구축된 환경은 <그림 2>와 같다.

4.2 실험 방법

소프트웨어가 운영체제에서 제거될 경우 하드디스크에 제거되지 않고 남아있는 파일들을 확인하기 위해 크게 세 가지의 시스템 상태로 디스크를 덤프하여 실험을 진행하였다.

- Pure State : 윈도우즈 운영체제에 아무것도 설치하지 않은 초기 상태
- Installed State: 윈도우즈 운영체제에 사용자가 소프트웨어 하나를 설치한 상태
- Removed State: 윈도우즈 운영체제에서 사용자가 설치한 소프트웨어를 제거한 상태



<그림 2> 스토리지에 남겨진 더미파일의 정보를 추출하기 위한 실험환경 구축

세 가지 상태에서 각각 디스크를 덤프하여 생성된 덤프 파일에서 운영체제 설치되어 있던 파일들의 리스트를 추출한 뒤 이를 비교하여 소프트웨어가 제거된 후에 남겨진 파일들의 개수와 크기를 확인한다. 남겨진 파일들의 개수와 크기를 확인하는 방법은 수식과 같다.

$$software \in staled File = \in staled State (file) - Pure State (file)$$

$$Removed File = Software \in staled File \cap Removed State (file)$$

설치된 소프트웨어의 설치 파일들을 추출하기 위해 Installed State에서 추출한 파일과 Pure State에서 추출한 파일 사이의 차집합 연산을 통해 소프트웨어 설치 파일들을 추출한다. 그 다음 추출한 소프트웨어 설치 파일 리스트(Software Installed File)과 소프트웨어가 제거된 상태 시점에 운영체제에 존재하는 파일(Removed State)의 교집합 연산을 사용하여 소프트웨어가 제거된 후에 하드디스크에 제거 되지 않고 남아있는 설치 파일들의 개수와 크기를 알아낸다. 또한 이러한 더미 파일들이 운영체제의 성능에 영향을 주는지 확인한다.

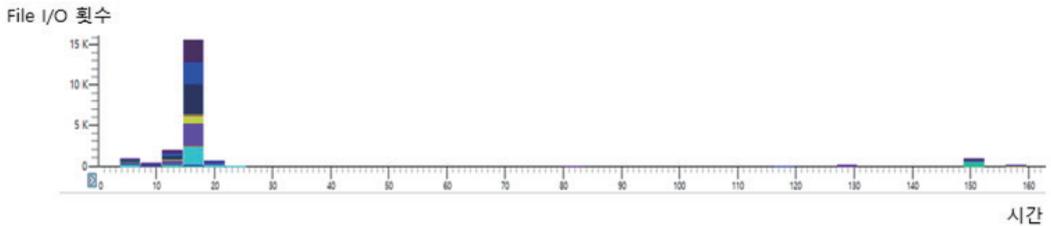
4.3 실험 결과

앞서 선정한 4개의 소프트웨어를 대상으로 위의 실험을 수행하였을 때 각각의 소프트웨어에 대해 제거되지않고 디스크에 남겨져 있는 파일들의 개수와 크기를 도출할 수 있었다. 실험에 대한 결과는 < 표 1>과 같다.

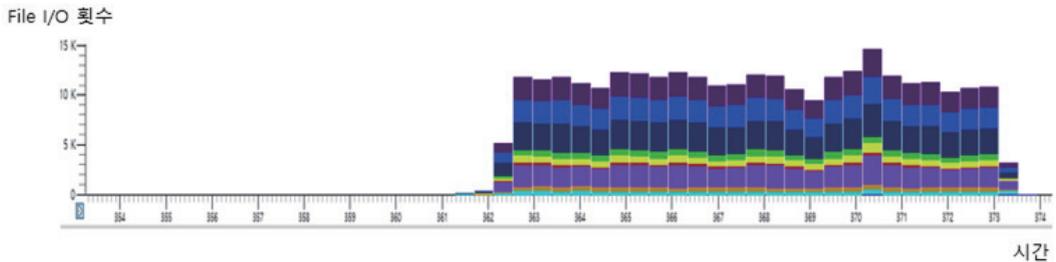
<표 1> 소프트웨어가 제거 되었을 때 디스크에 남겨진 파일 측정 결과

소프트웨어	지워지지 않은 파일의 개수 / 설치된 파일 개수	지워지지 않은 파일의 크기 / 설치된 파일 크기
Alyac	31.0%	63.0%
V3	62.6%	54.5%
KM Player	81.2%	71.8%
nProtect	78.9%	58.4%

소프트웨어가 운영체제에서 제거되었음에도 불구하고 디스크에 남아있는 파일은 로그 파일, 임시 파일, 설치 파일의 일부 또는 소프트웨어가 사용했던 파일 등이 있었다. 해당 실험을 반복하여 진행한 결과 약 5%의 오차가 있는 것을 확인할 수 있었다. 그럼에도 불구하고 소프트웨어가 제거된 후에도 설치파일의 절반 이상이 디스크에 제거되지않고 남아있는 것을 확인할 수 있다. <그림 4>는 소프트웨어가 시스템에서 제거되었음에도 불구하고 디스크에 저장되어있는 파일들의 리스트이다. 또한 개발된 도구를 사용하여 약 2년 사용한 시스템에서 이미 제거된 소프트웨어의 로그(.log) 파일 및 링크(.lnk) 파일 등 제거되지않은 더미 파일들을 추출하였을 때 약 3만 개가 하드디스크에 저장되어있는 것을 확인하였다. 일반적으로 소프트웨어가 시스템에서 제거될 경우 대부분의 임시 파일, 로그 파일, 링크 파일을 시스템에서 제거하지 않는다.



<그림 3> 운영체제에 더미파일이 존재하지 않을 경우의 시스템 File I/O 결과



<그림 4> 운영체제에 더미파일이 존재할 경우의 시스템 File I/O 결과

윈도우즈 운영체제를 새로 구축한 뒤 소프트웨어가 디스크에 남기고 제거되는 파일들이 시스템의 성능에 영향을 주는지 실험하였다. 시스템의 성능측정을 위해 윈도우즈 운영체제에서 제공하는 Windows Performance Tool을 사용하여 5분 동안의 File I/O를 모니터링 하였다. 실험 결과는 <그림 3>, <그림 4>와 같다. x축은 시간, y축은 File I/O의 발생 횟수이다. 각각의 그래프 색이 다른 이유는 서로 다른 File I/O를 위해 시스템 콜 이벤트를 발생시켰기 때문이다. 설치되어있는 소프트웨어가 존재하지 않음에도 불구하고 더미 파일이 있음으로서 지속적으로 파일 입출력 이벤트가 발생하였다. <표 2>와 같이 시간이 지남에 따라 개수에 따라 점차적으로 메모리 사용량을 증가시키고, 결론적으로 시스템의 성능을 저하시키는 것을 확인할 수 있었다.

<표 2> 더미파일의 개수에 따른 시스템 메모리 사용량

더미파일의 개수	메모리 사용량(Max: 4GB)
10,000	2217MB
20,000	2474MB
30,000	2812MB
100,000	3482MB

4. 결론

본 논문에서는 소프트웨어의 설치 및 제거 작업이 시스템 성능 저하의 원인이 될 수 있음을 증명하였다. 이를 증명하기 위해 스토리지 데이터 기반의 에이징 측정 도구를 개발하여 운영체제에서 소프트웨어가 설치되고 제거될 경우 운영체제는 설치된 소프트웨어에 대한 완전 제거를 보장하지 못한다는 결과를 도출하였다. 국내 소프트웨어들 중 사용자 많이 설치하는 4개의 소프트웨어를 대상으로 실험한 결과 소프트웨어가 설치되고 제거될 때 많은 파일들을 디스크에 남기고 제거되는 것을 실험을 통해 확인하였으며 디스크에 제거되지 않고 남겨진 파일들은 거의 설치 프로그램의 절반 이상의 크기를 차지하고 있음을 확인할 수 있었다. 해당 남겨진 더미 파일들을 새로 구축한 윈도우즈 시스템에 복사하였을 때, 어떠한 소프트웨어도 설치되지 않았음에도 불구하고 파일 입출력 작업이 지속적으로 발생하는 것을 확인할 수 있었으며 이로 인해 메모리 사용량 또한 급격히 증가하여 시스템의 성능저하가 발생하였다. 추후 연구에서는 본 논문에서 다루었던 하드 디스크에 지워지지 않고 남겨진 더미 파일 이외에 운영체제에서 소프트웨어가 설치되고 제거되는 작업이 발생했을 때, 소프트웨어가 설치 전과 제거된 후의 시스템의 레지스트리, 메모리 등의 전반적인 차이점을 비교하여 제거되었음에도 복구되지 않은 시스템 요소들이 시스템의 성능에 영향을 줄 수 있는지 확인한다.

참고문헌

장은태 (2020). 스토리지 데이터 기반의 시스템 에이징 측정 도구.

<https://github.com/zj3t/kingpc/blob/master/solver.py>

D. Cotroneo, R. Natella, R. Pietrantuono & S. Russo (2010). Software aging analysis of the linux operating system. 2010 IEEE 21st International Symposium on Software Reliability Engineering, IEEE.

S. Garg, A. van Moorsel, K. Vaidyanathan & K. S. Trivedi (1998). A methodology for detection and estimation of software aging. Proceedings Ninth International Symposium on Software Reliability Engineering (Cat. No. 98TB100257), IEEE.

M. Grottke, L. Li, K. Vaidyanathan & K. S. Trivedi (2006). Analysis of software aging in a web server. IEEE Transactions on reliability, 55(3), 411-420.

Kourai, Kenichi & Shigeru Chiba (2007). A fast rejuvenation technique for server consolidation with virtual machines. 37th annual IEEE/IFIP international conference on dependable systems and networks (DSN'07), IEEE.

Silva, Luis, Henrique Madeira & Joao Gabriel Silva (2006). Software Aging and Rejuvenation in a SOAP-based Server. Fifth IEEE International Symposium on Network Computing and Applications (NCA'06), IEEE.

국한문 참고문헌의 영문 표기

(English translation / Romanization of reference originally written in Korean)

Jang, Eun-Tae (2020). System aging measurement tool based on storage data.

<https://github.com/zj3t/kingpc/blob/master/solver.py>