

# uPC player: 윈도우 운영체제에서의 OS가상화 기반 무상태 컴퓨팅 실행 환경 지원 기술

(uPC player: An OS

## Virtualization-based Technique to Support Stateless Computing Execution Environment on Windows)

성 백 재 <sup>†</sup>

(Baekjae Sung)

박 찬 익 <sup>\*\*</sup>

(Chanik Park)

**요 약** Stateless computing[1,2]은 컴퓨팅 환경의 이동성을 용이하게 지원하는 기술로써, 여러 사용자들이 컴퓨팅 자원을 공유하는 환경에서 각 사용자 별 보안성을 확보하는 주요 기술로 자리잡고 있다. 또한 최근 가상화 기술의 확산 및 cloud computing의 발전에 따라, 사용자가 사용하는 설정과 데이터들(state)을 원격의 서버등과 같은 임의의 저장장치에 저장 후, 향후 어떠한 환경에서든 저장된 내용을 복원하여 사용자의 computing 환경을 이어가는 개념을 지원하기 위한 핵심적인 요소이다.

본 논문에서 제안하는 uPC player는 사용자에게 Windows 운영체제 환경에서의 Stateless computing 실행환경을 지원하는 기술로써, uPC player는 OS 기반 가상화 모듈인 uPC[3,4]를 기반으로 사용자에게 Windows 운영체제 컴퓨팅 환경을 지원한다. 본 논문에서는 Stateless computing

실행환경을 지원하기 위해 uPC player를 어떻게 설계하고 구현하였는지 설명한다. uPC player는 Windows desktop 환경과 연동하여 호스트 시스템의 실행환경과 uPC 가상 실행환경을 자유롭게 이동 가능하며, OS 기반 가상화 모듈에 기반함으로써 uPC player를 이용한 사용자 실행환경을 지원하는 데 소요되는 시간이 약 1초 정도이다.

**키워드** : 개인컴퓨팅환경, 운영체제계층 가상화, Stateless Computing, 유비쿼터스

**Abstract** Stateless computing supports a mobility of computing environment easily. It is becoming a major technology for securing personal user's information on shared computing environment. With the advance of virtualization technology and cloud computing, stateless computing is an essential part of personal computing environment connectivity (user's setting and data is stored in remote server or some storage, and it can be restored at any computing environment)

In this paper, we propose uPC player that supports stateless computing execution environment on Windows. uPC player provides Windows operating system to user by using an uPC OS virtualization module. In this paper, we leverage how uPC player is designed and implemented for supporting a stateless computing execution environment. uPC player provides a desktop switch between host-system execution environment and uPC virtual execution environment. And it needs just one second for loading uPC virtual execution environment by using OS virtualization-based technique.

**Key words** : Personal Workspace, OS-level Virtualization, Stateless Computing, Ubiquitous

### 1. 서 론

Stateless computing[1,2]은 컴퓨팅 환경의 이동성을 용이하게 지원하는 기술로써, 여러 사용자들이 컴퓨팅 자원을 공유하는 환경에서 각 사용자 별 보안성을 확보하는 주요 기술로 자리잡고 있다. 또한 최근 가상화 기술의 확산 및 cloud computing의 발전에 따라, 사용자가 사용하는 설정과 데이터들(state)을 원격의 서버등과 같은 임의의 저장장치에 저장 후, 향후 어떠한 환경에서든 저장된 내용을 복원하여 사용자의 computing 환경을 이어가는 개념을 지원하기 위한 핵심적인 요소이다. 쉬운 예는 웹브라우저를 통해 중앙 서버에서 제공하는 어플리케이션을 사용하는 것으로써, 이러한 경우 웹브라우저가 제공되는 어떠한 환경에서든 이전 작업하던 환경을 그대로 사용할 수 있다는 장점을 가진다. 본 논문에서는 단순한 어플리케이션이 아닌 실행환경(또는 데스크탑 환경)을 저장하는 기술을 사용하고 있으며, 이 경우 그 유용성이 더욱 증대된다.

가상화 기술은 최근 정보통신 분야에서 주목을 받고

· 본 연구는 논문은 2009년도 두뇌한국21사업과 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의(NIPA-2009-C1090-0902-0045) 연구결과로 수행되었음

· 이 논문은 2009 한국컴퓨터종합학술대회에서 'uPC player: Windows 운영체제에서의 OS가상화 기반 Stateless Computing 실행 환경 지원 기술'의 제목으로 발표된 논문을 확장한 것임

<sup>†</sup> 학생회원 : 포항공과대학교 컴퓨터공학과  
jays@postech.ac.kr

<sup>\*\*</sup> 종신회원 : 포항공과대학교 컴퓨터공학과 교수  
cipark@postech.ac.kr

논문접수 : 2009년 8월 13일

심사완료 : 2009년 11월 6일

Copyright©2010 한국정보과학회: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 받고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제16권 제1호(2010.1)

있는 기술로써, 가상화된 하드웨어를 제공하는 하드웨어 가상화 기술[5,6]과 가상화된 운영체제를 제공하는 운영체제 가상화 기술[7,8] 등으로 나눌 수 있다. 운영체제 가상화 기술의 핵심은 namespace forwarding으로써, 운영체제를 사용하는 응용프로그램 또는 서버에게 가상화(또는 namespace forwarding)된 운영체제의 자원들을 제공하는 기술이다. 이를 통해 사용자는 현재 설치된 호스트 운영체제를 기반으로 하는 복수개의 운영체제를 사용하는 효과를 가지게 된다. 운영체제 가상화 기술은 하드웨어 가상화 기술과 비교하여 크게 두 가지의 장점을 가진다. 첫 번째는 서버 등의 응용프로그램을 구동시키기 위해 가상머신(또는 가상실행환경)을 로딩 시키는 데 소요되는 시간이 월등하게 짧다는 점이다 - 하드웨어 가상화 기술의 경우 운영체제 부팅을 해야 함. 두 번째는 가상머신 개수의 확장성에서 이득을 가진다 - 하드웨어 가상화 기술의 경우 가상머신 개수의 운영체제가 하드웨어 자원을 점유하고 있음. 이러한 장점들을 기반으로 운영체제 가상화 기술은 서버 가상화 기술[9], 모바일 데스크탑 환경 지원 기술[10,11] 등으로써 주목 받고 있다.

본 논문에서 제안하는 uPC player는 사용자에게 Windows 운영체제 환경에서의 Stateless computing 실행환경을 지원하는 기술로써, uPC player는 OS 기반 가상화 모듈인 uPC[3,4]를 기반으로 사용자에게 Windows 운영체제 컴퓨팅 환경을 지원한다. 즉 uPC player에서 uPC 이미지(파일로 저장된 가상실행환경)를 로드하고, 이후 가상실행환경에서 변경된 내용은 바로 uPC 이미지로 저장된다. uPC 이미지는 원격의 서버, 로컬 저장장치, 휴대용 저장장치 등에 저장될 수 있으며, Windows 운영체제가 설치된 어떠한 환경에서도 uPC player를 통해 로드가 가능하다. uPC player는 Windows desktop 환경과 연동하여 호스트 시스템의 실행환경과 uPC 가상 실행환경을 자유롭게 이동 가능하며, OS 기반 가상화 모듈에 기반함으로써 uPC player를 이용한 사용자 실행환경을 지원하는 데 소요되는 시간이 약 1초 정도이다.

이후 2장에서는 이전 연구에 대해 간략히 소개하고, 3장에서는 Stateless computing을 위한 uPC player의 설계 및 구현에 대해 기술한다. 4장에서는 uPC 이미지 구동 오버헤드 성능 측정된 것을 보인다. 이후 5장에서는 결론과 함께, 향후 과제에 대해 소개한다.

2. Related Work

uPC Player에서 Stateless computing 실행 환경을 구성하기 위하여, 이전 연구인 uPC 기술을 사용하고 있다. 본 절에서는 uPC 기술에 대해 간략히 소개한다.

ubiquitous Personal Computing Environment (uPC)

[3,4]는 포항공대에서 2007년 Windows 운영체제 환경에서 개발된 OS 기반 가상화 모듈로써 모바일 데스크탑 환경을 지원한다. 이 기술을 통하여, 사용자는 자신의 데스크탑 환경(또는 컴퓨팅 환경)을 휴대용 저장장치에 저장하여, 자신이 원하는 어떠한 컴퓨터에서도 사용 가능하다.

uPC의 운영체제 가상화 기술의 핵심은, namespace forwarding이다. 가상실행환경 위의 응용프로그램의 운영체제 자원의 사용 요청은 namespace가 변경되어 요청이 내려가게 된다. 예를 들어 가상실행환경(Virtual PC) 상 응용프로그램의 파일시스템 자원 요청 C:\a.dat는 D:\a.dat로 namespace가 변경된다. namespace forwarding 기법은 커널 영역에 시스템콜 hooking을 통해 가능하다. uPC Virtualization Layer는 커널 디바이스 드라이버 형태로 로드되어, 원하는 운영체제 자원 사용 요청(시스템 콜)을 확인하고, 필요 시 namespace를 변경한다. 이를 통해 가상실행환경 위의 응용프로그램은 호스트 운영체제를 사용하는 것과 동일하게(또는 인식없이) 가상화된 운영체제를 사용하게 된다. 그림 1은 uPC 운영체제 가상화 기술을 전체적으로 도식화 하고 있다.

현재 uPC는 모바일 데스크탑 환경 지원이라는 디자인에 따라, Filesystem, Registry, Kernel Object, Service 자원을 가상화 해주고 있으며, 휴대용 저장장치 외에도 유·무선 네트워크 저장장치 등을 활용한 시나리오를 통해 그 유용성을 검증/확장 하고 있다. 자세한 내용은 참고문헌[3,4]을 참고바란다.

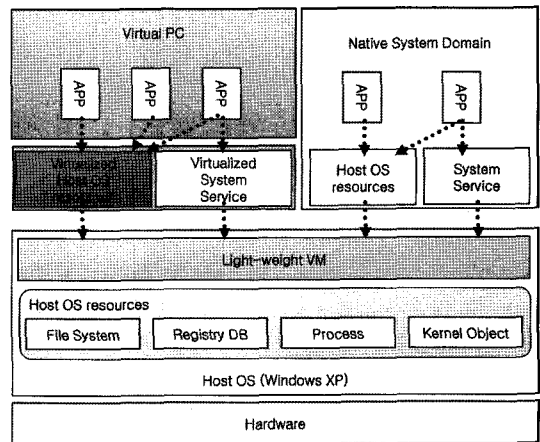


그림 1 uPC 운영체제 가상화 기술

3. u-PC

uPC player는 기존 uPC 운영체제 가상화 모듈을 이용하여, Stateless computing 실행 환경을 지원하는 기술이다. uPC player는 uPC 이미지(저장된 가상실행환

경 파일)를 열고, 가상실행환경을 위한 환경을 설정하고, 가상실행환경을 로드한다. 자세한 내용은 세부항목에서 설명한다.

uPC player는 Stateless computing 실행 환경 지원을 위해 세가지 항목에 초점을 맞추어 설계 및 구현되었다. 첫째, uPC 가상실행환경에서 변경된 모든 사항은 바로 uPC 이미지로 저장되어야 한다 - 일반적으로 uPC 이미지는 원격 서버 또는 휴대용 저장장치에 존재한다. 둘째, uPC 가상실행환경이 사용된 이후, 호스트 운영체제에는 어떠한 변경도 있어선 안 된다. 셋째, 저장된 uPC 이미지는 어떠한 곳에서도 실행가능 해야 한다 - 본 논문에서는 Windows XP가 설치된 환경이라고 가정한다.

**3.1 Stateless 컴퓨팅 환경 지원**

본 항목을 위하여 uPC player에서는 두 가지 측면에서 지원해 주고 있다.

첫 번째는 uPC 가상실행환경 로드/언로드 시점에 사용되었던 가상화된 filesystem, registry, kernel object, service 자원에 대한 처리를 해주는 것이다. 예를 들어, uPC player는 가상 registry 자원을 제공하기 위하여, registry hive 파일을 호스트 registry hierarchy의 sub-tree에 로드/언로드한다. 이 과정은 정상적으로 처리가 이루어져야 한다. 또한 가상 kernel object 자원을 제공하기 위한, uPC의 kernel object directory는 정상적으로 생성/삭제시켜야 할 것이다. 부가적으로 namespace forwarding을 위한 uPC virtualization layer의 시스템콜 루틴에서 호스트 운영체제에 변경이 없도록 robust하게 처리를 해주어야 할 것이다.

두 번째는 uPC만의 새로운 desktop object를 생성하여 사용하는 것이다[12]. desktop object는 kernel object로써, windows, menus, hooks와 같은 User Interface Object에 관한 정보들을 가지고 있다. 따라서 새로운 desktop object를 사용하여 가상실행환경을 로드 할 경우, desktop 전환(switch)에 따라 user profile (예: 프로그램목록, 바탕화면이미지 등), tasklist(활성&비활성화 되어있는 프로그램 목록) 등이 전환하게 된다 - default desktop, uPC(새로운) desktop 간 전환 시, 이는 사용자가 다른 운영체제(또는 실행환경)를 사용하는 것처럼 보이는데 적합하며, tasklist 등이 별도로 유지되므로 호스트 운영체제에 변경을 하지 않는 측면에서 장점을 가진다. 또한 desktop object의 특징으로, 동일한 desktop 내의 프로세스간에만 message가 전송 가능하므로 가상실행환경 간의 isolation 측면에서도 장점을 가진다. 그림 2는 새로운 uPC desktop object를 생성하고 해당 desktop object를 기반으로 하는 응용프로그램 실행/사용하는 디자인을 보여준다.

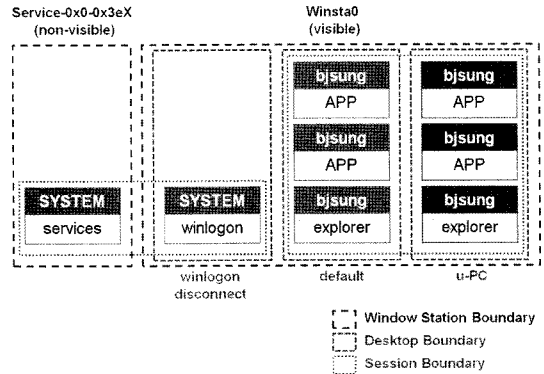


그림 2 uPC desktop object 추가

**3.2 uPC 가상 실행환경의 이동성 지원**

uPC player는 저장장치에 uPC 이미지(저장된 uPC 가상실행환경)를 저장/복원해주며, 이 저장장치는 유무선 네트워크 스토리지, 로컬 저장장치, 휴대용 저장장치 등의 어떠한 저장장치라도 사용이 가능하다. 이 특징은 기존 uPC 기술에 기반한다. 즉 uPC의 운영체제 가상화 기술은 일반적인 운영체제 가상화 기술(예: FVM [7] 등)과 다르게 namespace forwarding을 해주고 있으며, 이러한 특징은 uPC 이미지가 Windows가 설치된 어떠한 host PC에서도 uPC 가상실행환경이 수행 가능하도록 해 준다. 이 내용을 요약하여 설명하면 다음과 같다.

일반적인 운영체제 가상화기술은 단순히 filesystem과 registry의 전체 namespace에 대하여 Copy-On-Write 정책으로 처리한다. 하지만 u-PC의 경우 특정 namespace는 host PC의 자원과 완전히 분리시켜 host PC의 자원을 전혀 사용하지 않는다. 이럴 경우, uPC 이미지는 host PC의 자원의 상태에 의존하지 않고, 가상실행환경을 로드하고 사용할 수 있게 된다. 예를 들어, host PC에는 이미 Word 프로그램이 설치되어 있고, 가상실행환경 위에서 동일한 Word 프로그램을 설치한다고 가정해 보자. 만약 filesystem과 registry의 전체 namespace에 대하여 Copy-On-Write 정책을 사용한다면, 설치 프로그램은 host PC의 namespace를 읽고, 설치할 파일이 이미 존재함을 인식 후 해당 파일들을 저장하지 않을 것이다. 즉 Word를 실행시키기 위한 모든 file과 registry는 uPC 이미지에 저장되지 않을 것이다. 이러한 문제점 등을 해결하기 위해서 uPC 운영체제 가상화 기술은 설계되었으며, namespace forwarding 정책을 다음과 같이 두고 있다. 먼저 filesystem 자원과 관련된 정책은, 개인설정 및 프로파일과 관련된 Documents and Settings 디렉토리와 응용프로그램 설치 경로인 Program Files 디렉토리는 무조건 uPC 이미지가 있는 namespace로 forwarding한다. 윈도우 운영체제

필수파일들과 관련된 WINDOWS 디렉토리는 기존 Copy-On-Write의 정책을 그대로 사용한다. 또한 registry 자원과 관련된 정책은, \REGISTRY\MACHINE 하위의 HARDWARE, SAM, SECURITY 키를 제외한 모든 키들은 무조건 uPC 이미지가 있는 namespace로 forwarding한다. 이에 대한 자세한 내용은 이전 논문[3,4]을 참조하기 바란다.

#### 4. 성능 측정

본 절에서는 uPC player에서 Stateless computing 실행환경을 구성하기 위하여, uPC 이미지(저장된 가상 실행환경 파일)를 사용하여 가상실행환경을 구성하는데 소요되는 오버헤드를 측정한다. 본 논문에서는 uPC player의 성능 측정에만 초점을 맞추어 설명한다. uPC 기술의 가상화 오버헤드에 관한 내용은 이전 논문[3,4]을 참고하기 바란다.

실험환경은 다음과 같다. host PC는 Intel Core2Duo E6600(2.4 GHz) CPU, 4GB RAM, 400GB SATA-II disk(7200 RPM)를 사용하였으며, 운영체제로 Windows XP SP3를 사용하여 성능을 측정하였다.

##### 4.1 uPC 가상실행환경 로딩 오버헤드

표 1은 uPC player에서 uPC 이미지(저장된 가상실행환경 파일)를 사용하여 가상실행환경을 구성하는데 소요되는 시간을 각 작업에 따라 세부항목으로 나누어 보여주고 있다. 먼저 “실행환경 환경 정보 수집” 작업시간은 현재 가상실행환경 실행을 위한 정보, 즉 SYSTEM-DRIVE 또는 현재 사용자 SID와 같은 정보를 수집하는데 소요되는 시간을 의미한다. “uPC registry hive 로드”는 3.1절에 설명한 registry hive를 로드하여 uPC registry hierarchy를 구성하는데 소요되는 시간이며, “uPC kernel driver 로드”는 시스템콜을 hook하는 uPC virtualization layer를 구성하는데 소요되는 시간이다. 또한 “uPC desktop 생성”은 desktop객체를 생성하는 시간이며, “uPC 가상실행환경 구성”은 uPC 가상실행환경의 사용을 위해 필요한 uPC desktop내에 explorer.exe 실행 등의 작업에 소요되는 시간을 보여준다. registry hive를 로드하는데 가장 많은 시간이 소요되는 것을 볼 수 있으며, uPC kernel driver를 로드하는 시

표 1 uPC 가상실행환경 구성 소요시간

작업	소요시간 (ms)
실행환경 환경 정보 수집	0.471
uPC registry hive 로드	735.756
uPC kernel driver 로드	269.556
uPC desktop 생성	0.0155
uPC 가상실행환경 구성	10.111
Total	1015.909

간도 꽤 큰 비중을 차지하는 것을 볼 수 있다. 하지만 전체소요되는 시간이 대략 1초 정도로 uPC 가상실행환경을 구성 오버헤드는 비교적 크지 않음을 볼 수 있다.

#### 5. 결론 및 향후 연구

지금까지 uPC player가 어떻게 Stateless computing 실행환경을 지원하는지에 대해 설명하였다. uPC player는 기존 uPC의 운영체제 가상화 기술을 사용하여 가상 실행환경을 제공해주고 있다. 이를 통해, 가상실행환경에서 변경된 모든 사항은 바로 uPC 이미지로 저장되고 있으며, 또한 가상실행환경이 사용된 이후 호스트 운영체제에는 어떠한 변경도 발생하지 않음을 보였다. 그리고 저장된 uPC 이미지는 어떠한 곳에서라도 실행 가능하도록 하기 위해, uPC 운영체제 가상화는 일반적인 운영체제 가상화와 다르게 namespace forwarding을 처리함을 보였다.

또한 uPC player가 uPC 가상실행환경을 구성하기 위하여, 어떠한 작업을 수행하는지 또한 어느 정도의 오버헤드가 발생하는지 성능측정 장에서 설명하였다. uPC 가상실행환경 로딩 오버헤드는 registry hive를 로딩하는데 가장 많은 시간이 소요되었지만, 전체 로딩시간이 대략 1초 정도로 짧은 시간에 가상실행환경이 로딩되는 것을 알 수 있다.

uPC player는 Stateless computing 실행환경을 지원한다는 점에서 그 활용도가 크다. 특히 운영체제 가상화 기술을 사용하여 가상실행환경을 구성함으로써, 기존 하드웨어 가상화 기술과 차별화가 되며, 가상실행환경 로딩오버헤드 및 가상실행환경 개수의 확장성 면에서 이점을 가지므로 그 활용할 수 있는 응용이 많다. 먼저 간단하게, 최근 주목 받고 있는 cloud computing에서 stateless computing을 사용하는 것과 유사한 개념으로, uPC 이미지를 원격의 서버에 저장해 두고 사용자는 Windows가 설치된 어떠한(또는 공용) PC에서 uPC player를 통해 이미지를 로드하여 실행환경을 구성하는 응용을 생각할 수 있다. 또한 하나의 PC에서 복수 개의 가상실행환경을 로드하여 복수 개의 서버를 수행하는 응용 등을 구성할 수 있다. 향후 uPC player의 유용성을 검증하기 위해, 활용방안 및 구체화된 응용 시나리오를 연구할 예정이다.

#### 참고 문헌

- [1] Ryan Paul, "Stateless computing: the future of the cloud?," <http://arstechnica.com/hardware/news/2008/08/stateless-computing-the-future-of-the-cloud.ars>.
- [2] Jeff Fisher, "The state of stateless computing," <http://www.desktopsasaservice.com/blog/2008/5/7/the-state-of-stateless-computing.html>.

- [3] I. Kim, M. Hwang, W. Lee, C. Park, "u-PC: Personal Workspace on a Portable Storage," The 4th Int'l Conf. Mobile Technology, Application and Systems (Mobility Conf. 2007), Singapore Chapter for ACM, pp.228-233, Sept. 2007.
- [4] B. Sung, I. Kim, M. Hwang, W. Lee, C. Park "Wireless u-PC: Personal workspace on an Wireless Network Storage," *Journal of KIISE : Computing Practices and Letters*, vol.14, no.9, pp.916-920, Dec. 2008. (in Korean)
- [5] VMWare Pocket ACE, <http://www.vmware.com/products/ace/>.
- [6] Citrix XenDesktop, <http://www.citrix.com/english/ps2/products/product.asp?contentID=163057>.
- [7] Yang Yu et al, "A Feather-weight Virtual Machine for Windows Applications," *Second Int'l Conf. Virtual Execution Environments (VEE 06)*, ACM, pp.24-34, June 2006.
- [8] RingCube mojopac, <http://www.mojopac.com/>.
- [9] Parallels Virtuozzo, <http://www.parallels.com/virtualization/server/>.
- [10] U3, <http://www.u3.com/>.
- [11] Ceedo, <http://www.ceedo.com/>.
- [12] Malli. S, "Virtual Desktop: A Simple Desktop Management Tool," <http://www.codeproject.com/KB/system/VirtualDesktop.aspx>.