



한국정보과학회  
KOREAN INSTITUTE OF INFORMATION SCIENTISTS AND ENGINEERS  
<http://www.kiise.or.kr>

# 제41회 한국정보과학회 정기총회 및 동계학술발표회

2014년 12월 18일(목) ~ 20일(토), 강원도 휘닉스 파크

<http://www.kiise.or.kr/conference02/>

3D Printing Workshop 12.18(목)

슈퍼컴퓨팅 최근 연구 동향 워크샵 12.18(목)

사물인터넷연구회 창립총회 및 워크샵 12.18(목)

모바일과 사물인터넷의 만남 12.18(목)

데이터베이스 연구동향 워크샵 12.19(금)

|후원|

골드

SAMSUNG

삼성전자



LG전자

실버

NAVER

SAMSUNG

삼성SDS



# 2014 동계학술대회 논문집

2014년 12월 18일(목) ~ 20일(토), 강원도 휘닉스 파크



40. 클라우드 컴퓨팅을 이용한 고품질 멀티미디어 시스템의 설계 및 구현 .....	문종배 · 조정현 · 김성운 · 김학영	1159
41. 사이버 물리 시스템을 위한 가변 수행 시간 태스크 모델 지원 실시간 시뮬레이션 기법 .....	위경수 · 김승곤 · 이창건	1162
42. 임베디드 시스템의 온 더 플라이 동적 재프로그래밍을 위한 시스템 설계 방식 .....	김승곤 · 이창건	1165
43. [우수논문] 로그 인지 플래시 변환 계층 .....	김동욱 · 강수용	1167
44. 인피니밴드 스토리지 네트워크를 적용한 오픈스택 클라우드 스토리지 시스템의 설계 .....	허희성 · 이광수 · 매히디 · 김덕환	1170
45. 고장 예측 시스템 검증을 위한 센서 신호 생성기 .....	탁해성 · 박보국 · 조환규	1173
46. 단어-거리 그래프를 이용한 주제어 추출 시스템 구현 .....	탁해성 · 조환규	1176
47. 가상화 환경에서의 스토리지 I/O 패턴 분석 .....	오찬수 · 강동현 · 엄영익	1179
48. 셀프-힌트를 이용한 가상머신 메모리 공유 기회 향상 기법 .....	남예지 · 이동우 · 엄영익	1181
49. 계층적 스케줄링 시스템의 스케줄링 가능성 검증을 위한 UPPAAL 모델 .....	안소진 · 황대연 · 최진영	1184
50. 이동 에이전트를 이용한 약조정 검사점 설정 기법 .....	박태순	1187
51. 리눅스 CFS 그룹 스케줄링을 위한 엄격한 CPU 자원예약 .....	임인구 · 조현철 · 진현욱 · 이상일	1190
52. 메모리 집약적 환경에서 분할 전송 및 더티 페이지 중복 제거 방식을 이용한 가상머신 라이브 마이그레이션 수행시간 감소 기법 .....	차재근 · 신치훈 · 김학영	1193
53. Nearest Neighbor 알고리즘 기반 MapReduce 자원 할당 구조 제안 .....	홍승길 · 이민수 · 오상윤	1196
54. 작은 기기를 위한 저메모리 자바스크립트 엔진의 최소 표준규격 연구 .....	김민수 · 정혁진 · 문수목	1199
55. 스프레드시트 데이터를 이용한 시각적 FMEA 분석 시스템 구현 .....	이재호 · 최지수 · 김상철 · 남종호	1202
56. PCM을 활용한 가상화 환경에서의 페이지 공유 기법 .....	이민호 · 이동우 · 엄영익	1205
57. [우수논문] 페이지 캐시를 위한 바이트 단위의 PCM 성능 분석 .....	강동현 · 엄영익	1208
58. IOV 기반 가상 데스크탑 서비스를 이용한 물리적 네트워크 망분리 시스템 설계 및 구현 .....	김선욱 · 김성운 · 김학영 · 정성권 · 이숙영	1210
59. IoT 가속도 센서 모듈 하드웨어 설계 .....	임호정 · 강정훈 · 지영민 · 유준재	1213
60. IoT 가속도 센서 소프트웨어 설계 .....	임호정 · 강정훈 · 지영민 · 유준재	1215
61. 진동 센서 모니터링 연구 .....	임호정 · 강정훈 · 지영민 · 유준재	1217

## 컴퓨터이론

1. $\alpha$ -Chordal 그래프 색칠하기 .....	이춘석	1219
2. 경량화된 유전알고리즘 기반 순회판매원 문제 해결기법 .....	노태완 · 오하영 · 김중권	1222
3. 해싱기반 순위다중패턴매칭 알고리즘 .....	강문성 · 조석현 · 심정섭	1225
4. Snort에서 편집거리를 이용하여 탐지규칙을 추가하고 분류하는 도구 개발 .....	김혜미 · 조호성 · 박희진	1228

# 셀프-힌트를 이용한 가상머신 메모리 공유 기회 향상 기법

남예지, 이동우, 엄영익

성균관대학교 정보통신대학

{yejinam, lightof, yieom}@skku.edu

## Improving Opportunity for Memory Sharing using VM Self-Hint

Yeji Nam, Dongwoo Lee, and YoungIk Eom

College of Information and Communication Engineering., Sungkyunkwan Univ.

### 요 약

현재 대부분의 데이터 센터에서는 소모 전력 감소와 공간 확보를 위해 가상화를 이용한 서버통합을 수행하고 있다. 이러한 가상화 서버에는 서버 통합의 효율성을 높이기 위해 메모리 공유기법이 널리 사용되고 있다. 그러나 기존 공유 기법은 메모리 전체의 내용을 확인하는데 따른 성능 저하가 발생하며, 짧은 시간만 존재하는 공유 기회를 확인하지 못하는 문제가 있었다. 본 논문에서는 게스트에서 직접 호스트에 공유 가능성이 높은 메모리 힌트를 전달하여 메모리 공유 기회를 향상시킬 수 있는 공유 기법을 제안한다. 실험을 통해 게스트의 공유 가능성이 높은 시스템 디렉토리에 속한 파일 정보를 전달함으로써 공유 기회를 높일 수 있음을 확인하였다.

### 1. 서 론

가상화는 더 이상 우리에게 낯선 기술이 아니다. 거의 대부분의 데이터 센터에서 공간 제약 해결 및 소모전력 감소를 위해 가상화 기술을 통한 서버의 통합이 이루어지고 있다. 높은 수준의 서버통합을 위해서는 하나의 물리머신에 동작하는 가상머신의 수를 늘리는 것이 가장 중요하므로 효율적인 자원분배가 가상화 기술의 핵심이라고 할 수 있다. 프로세서 자원 및 디스크와 NIC같은 장치 자원은 각 가상머신에 시분할 방식으로 공유하는 것이 가능하다. 반면 각 가상머신이 필요로 하는 만큼 상호 배타적으로 할당해주어야 하는 메모리[1]는 통합 가능한 서버의 수를 제약하는 가장 큰 원인이다.

가상머신의 효율적 메모리 사용을 위한 여러 가지 방법이 연구되어 왔으며, 운영체제에서 널리 사용되는 COW(Copy-On-Write) 기반의 메모리 공유와 유사한 가상머신 메모리 공유 기법[2]이 많은 하이퍼바이저에서 사용되고 있다. 메모리 공유 기법의 핵심은 공유 가능한 메모리를 효율적으로 찾아내 제거하는 것이다. 가장 직관적이고 대중적인 방법은 메모리를 순차적으로 스캔해 나가며 메모리 내용을 비교하는 방법이다. 그러나 순차적 비교를 통해 공유 가능한 메모리를 찾는 것은 시스템에 큰 오버헤드를 유발하며 검색시간 간격에 발생하는 짧은 메모리 공유 기회를 놓치게 되는 단점이 있다. 이러한 단점을 해결하기 위한 몇 가지 연구들이 있었으나, 적용 가능한 환경의 제약이나 게스트-호스트 간의 정보 제약(Semantic Gap)으로 인한 효율성 저하의 문제점이 여전히 남아 있다.

본 논문에서는 최근 게스트-호스트 간의 시맨틱 갭을

줄이기 위해 널리 사용되는 게스트 모듈을 활용하여 공유 가능성이 높은 파일 정보를 게스트로부터 직접 전달받아 공유 가능한 메모리의 양을 늘리고 불필요한 중복 페이지 탐색으로 인한 성능 저하를 막을 수 있는 페이지 공유 기법을 제안한다. 또한 본 기법의 유효성을 평가하기 위해 여러 운영체제간의 파일 유사성을 비교하는 실험을 진행하여 윈도우 및 리눅스 운영체제의 버전에 관계없이 특정 시스템 폴더에 속한 파일에서 유사성이 나타남을 확인 하였다.

본 논문의 나머지 장은 다음과 같이 구성되어 있다. 2장에서 효율적 가상머신 메모리 관리를 위한 기술 및 관련 연구에 대해 소개한다. 3장에서는 본 논문에서 제안하는 새로운 공유 시스템에 대해 설명하고, 이 시스템이 적용될 경우 상당히 많은 공유 기회를 확보 할 수 있음을 4장에서 실험을 통해 확인한다. 마지막으로 5장에서는 본 논문의 결론을 서술한다.

### 2. 메모리 공유 기술 및 관련연구

시스템 구조 및 전력의 한계로 인해 실제 서버에 탑재 가능한 메모리는 제한되기 때문에 통합 가능한 서버의 수를 늘리기 위해서는 한정적인 메모리를 효율적으로 사용하는 것이 가장 중요하다. 이러한 방법은 크게 두 가지로 나눌 수 있다. 첫 번째는 각 가상머신이 필요로 하는 메모리를 예측하여 그에 맞는 메모리를 정확하게 분배해 메모리의 낭비를 막는 메모리 균형 분배기법[3]이며, 두 번째는 각 가상머신이 사용하는 메모리에서 중복된 부분을 찾아내어 해당 부분을 각 메모리가 공유하게 하는 방법이다. 이 기법은 중복을 찾는 단위에 페이지 프레임 사용하기 때문에 흔히 페이지 공유기법 이라고도 불린다. 균형기법과 공유기법은 상호 보완적인 기술이다. 예를 들어 균형기법을 통해 완벽하게 메모리가 분



배된 호스트라 하더라도 갑작스러운 메모리 요구량의 변화로 인해 메모리가 부족해진 가상머신에서 메모리 스왑이 발생해 큰 성능저하가 발생할 수 있으나, 공유 기법을 통해 여유 메모리를 확보해 둔 호스트의 경우 이러한 현상을 방지 할 수 있다.

기존의 많은 하이퍼바이저가 다양한 페이지 공유 기법을 적용하고 있다. 이중 가장 널리 쓰이는 방법은 전체 메모리의 내용을 비교하여 중복을 찾는 내용기반 페이지 공유(Content-based Page Sharing)기법[2,4]이다. 그러나 이 기법은 공유 가능한 페이지를 찾는 과정에서 발생하는 시스템 성능 저하를 최소화하기 위해 메모리 일부 영역을 순차적으로 탐색할 수밖에 없는 한계점이 있다. 이로 인해 공유 가능한 페이지가 시스템에 존재하더라도 해당 페이지가 메모리 탐색 간격보다 더 짧은 시간동안만 사용된다면 해당 영역에 대한 공유기회를 잃어버리게 되는 단점이 있다.

가상머신들이 같은 파일을 접근 하는 경우 공유 가능한 페이지가 각 가상머신의 버퍼 캐시에 존재하게 되는 '이러한 버퍼 캐시에 대부분의 공유 기회가 있는 것에 착안하여 가상머신의 가상 디스크 접근 정보를 통해 공유 페이지를 찾아내는 방법이 제안되었다. 그러나 Satori [5]는 같은 디스크 이미지를 사용하는 가상머신에만 적용할 수 있으며, XLH [6]는 가상머신과 호스트의 정보 제약으로 인해 공유 가능성이 거의 없는 파일의 페이지까지 처리하게 되는 문제점이 있다.

### 3. 셀프 힌트를 이용한 페이지 공유 기법

이번 장에서는 본 논문에서 제안하는 공유 시스템의 동작과정을 파일시스템 모니터링, 파일시스템정보 전달, 가상머신 간 데이터 공유의 세 가지 단계로 구분하여 각 단계의 핵심 내용을 설명한다.

#### 3.1 게스트 파일시스템 모니터링

가상머신 간 메모리 공유에서 가장 큰 비중을 차지하는 부분이 파일 데이터를 담고 있는 디스크 버퍼 캐시라는 것은 널리 알려진 사실이다. 파일 데이터 중에는 가상 머신을 사용하는 사용자의 개인 파일과 같이 중복 가능성이 낮은 파일도 있지만 라이브러리 및 시스템 파일 [1]과 같이 대부분의 공유 기회를 차지하는 파일도 있다. 그러므로 공유시스템에서 게스트가 접근하고 있는 파일 정보를 알아 낼 수 있다면 공유 기회가 낮은 개인 파일 데이터를 제외하고 공유 기회가 높은 데이터만을 처리함으로써 시스템 전체의 메모리 공유 비율을 높이고 불필요한 데이터 비교를 줄일 수 있을 것이다. 그러나 가상화 환경에서의 게스트-호스트의 분리로 인한 정보제약이 존재하여 이러한 정보 교환이 어려웠다.

본 논문에서 제안하는 공유 시스템은 이러한 정보제약을 완화시키기 위하여 게스트 내부에서 파일시스템 정보를 수집하여 호스트에게 직접 전달하는 모니터링 모듈을 추가 하였다. 게스트에 추가적인 모듈을 사용하는 것은 가상머신의 성능에 영향을 미칠 수 있으나, 이러한 추가적인 모듈을 사용해 모듈 동작에 의해 발생하는 오버헤드보다 더 큰 성능 및 동작 효율성 향상을 가져올 수 있음이 최근 여러 연구에서 확인 되었다.

모니터링 모듈은 일정 주기마다 그림 1과 같은 과정을 거쳐 파일시스템에서 사용 중인 파일의 정보를 수집/전달하게 된다. 우선 공유 가능성이 높은 파일의 정보만을

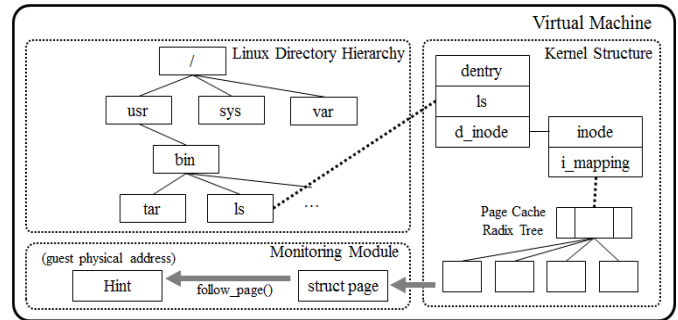


그림 1 모니터링 모듈의 파일시스템 정보 수집 과정

수집하기 위해 라이브러리나 시스템 파일이 저장된 디렉토리를 미리 지정한다. 이렇게 지정된 디렉토리의 포함된 파일들의 d\_entry를 차례로 순회하여 접근한다. 그 중 유효한 인덱스 노드를 가진 파일들에 대해서는 인덱스 노드의 i\_mapping 필드를 통해 해당 파일의 데이터를 페이지 프레임 단위로 수집한다. 수집된 페이지 프레임의 물리주소를 호스트에 전달하여 공유 시스템의 힌트로 활용한다.

#### 3.2 게스트-호스트 파일시스템 정보 전달

모니터링을 통해 얻어진 파일시스템 정보를 호스트에서 전달하는 과정에서 불필요한 데이터 복제가 일어나지 않도록 게스트와 호스트간의 공유 메모리를 이용한다. 게스트의 물리 메모리 공간은 호스트 입장에서는 프로세스 혹은 도메인의 가상 메모리 공간이므로, 게스트 내부에 일정 공간을 할당하고 이 공간의 물리주소를 호스트에 알려주는 것만으로 공유 메모리 구성은 간단히 이루어진다. 공유 메모리에 파일 시스템 정보를 담은 후 이를 호스트에 통지하는 것은 'vmexit'을 이용한다. 잦은 'vmexit'은 시스템 성능 저하의 한 원인이지만, 본 공유 시스템의 모니터링 주기는 컴퓨터 시스템의 동작시간에 비하면 매우 긴 시간(기본값 5초)이므로 CPU자원의 소모가 심한 폴링방법보다 더욱 효율적이다.

게스트에서 전달한 파일시스템 정보는 게스트의 물리 메모리 공간의 주소를 나타내고 있으므로, 이를 호스트에서 활용하기 위해서는 호스트의 공유시스템 모듈에서 접근 가능한 메모리 공간으로 변환해 주어야 한다. 앞에서 설명한 게스트-호스트간 공유 메모리와 마찬가지로 게스트의 물리주소공간은 호스트입장에서는 가상주소공간이므로 커널에서 제공하는 가상주소-물리주소 변환 메커니즘을 이용하면 해당 주소 변환이 가능하다. 이렇게 변환된 정보는 가상머신 간 데이터 공유를 위해 우선순위 스택(Priority Stack)이라는 특별한 자료구조에 저장된다. 해당 자료구조의 자세한 구조 및 동작은 다음의 3.3 절에서 설명한다.

#### 3.3 파일시스템 정보기반 가상머신 간 데이터 공유

호스트에 전달된 게스트의 파일 시스템 정보는 실제 가상머신 간 데이터 공유에 이용된다. KSM에서 사용되는 페이지 비교 및 공유 메커니즘을 이용해 공유 시스템을 구현함으로써 기존의 스캔 방식으로 얻어진 공유 메모리와 게스트의 파일 시스템 정보를 통해 얻어진 공유 메모리를 함께 관리 할 수 있게 된다. 과도한 메모리 공유시스템의 동작으로 인한 시스템 성능 저하를 막기 위해 본 시스템은 그림 2와 같이 KSM과 번갈아가며 동작한다. 각

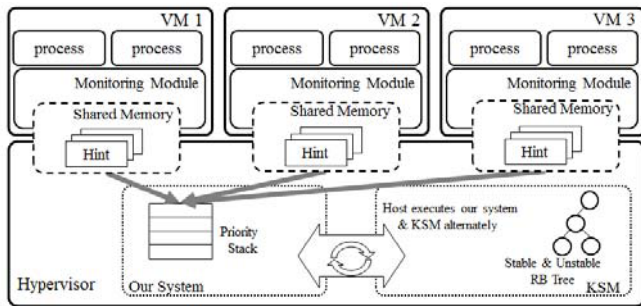


그림 2 Linux KSM을 활용한 공유 시스템의 동작 과정

스캔 타임마다 기존의 KSM과 본 공유 시스템 중 하나의 동작이 수행 되므로 이는 KSM의 스캔 간격을 2배로 증가시켜 짧은 공유 기회에 대한 공유 가능성을 더욱 잃게 되지만 공유 가능성이 높은 가상머신 내부의 파일 내용을 직접 다른 가상머신의 파일 내용과 비교하게 됨으로써 전체적으로는 공유 기회 향상을 가져오게 된다.

시간이 지남에 따라 게스트로부터 전달 받은 파일시스템 정보의 누적량은 점점 증가하게 된다. 이렇게 전달 받은 파일 시스템 정보는 게스트에서 더 이상 파일이 사용되지 않게 되는 등의 이유로 점차 그 유효성을 상실할 수 있다. 또한 늘어나는 파일시스템 정보를 모두 처리하려면 이로 인한 오버헤드도 크게 증가하게 된다. 이러한 문제를 방지하기 위해서는 적절한 수준으로 파일시스템 정보를 관리하는 것이 중요하다. 우리는 최소-힙(Min-Heap)과 유사한 우선순위 스택 자료구조를 사용해 이러한 문제를 해결하였다. 우선순위 스택은 구조적으로는 배열과 유사하기 때문에 적은 비용으로 순차적 접근이 가능하며, 불필요한 항목을 제거할 때는 '오래된 파일정보 일수록 우선순위가 낮음', '여러 가상머신이 동시에 접근하고 있는 디렉토리에 속한 파일정보 일수록 우선순위가 높음'과 같은 두 가지 우선순위 정책을 통해 우선순위가 낮은 파일 시스템 정보를 제거한다.

#### 4. 실험 및 평가

본 논문에서는 게스트에서 직접 접근 중인 파일 정보를 전달함으로써 공유 기회를 향상 시키고 불필요한 중복 검색 오버헤드를 감소시킬 수 있다고 하였다. 이러한 공유 기회 향상은 가상머신들이 공통으로 갖고 있는 시스템 디렉토리에 유사한 파일이 많을 경우 그 효용성이 극대화된다. 우리는 본 논문에서 제안하는 공유 시스템의 실효성을 평가하기 위해 실험을 통해 각 운영체제의 버전에 따라 시스템 디렉토리에서 어느 정도의 유사성이 나타나는지 확인하였다.

본 실험은 Windows 7과 8, 그리고 리눅스 배포판 중 Ubuntu 14.04, Ubuntu 14.10 및 Centos 6.4 와 같이 서로 다른 버전의 운영체제를 대상으로 수행 되었다. 실험은 각 운영체제의 초기 설치 상태에서 각각 많은 시스템 파일을 포함하고 있는 디렉토리를 선택하여, 각 운영체제 버전의 폴더에 있는 파일을 비교하는 과정으로 진행 하였다. 파일 비교는 단순히 이름과 크기를 비교하는 것이 아니라 각 파일의 내용까지 대조하여 완전히 같은 파일을 Identical, 서로 다른 버전에 존재하지만 그 내용이 일부 같은 파일을 Similar, 그리고 어느 한쪽의 버전에만 존재해 공유 가능성이 전혀 없는 파일을 Different로 구분해 각각의 파일의 개수를 측정하였다.

그림 2의 실험 결과를 자세히 살펴보면 Windows 7과

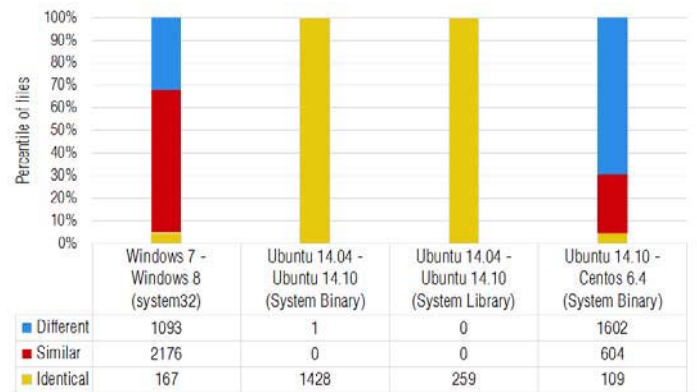


그림 3 리눅스 및 윈도우즈 운영체제의 시스템 디렉토리 파일 유사도 측정 실험 결과

8의 경우 완전히 같은 파일은 167개로 약 5%를 차지해 상대적으로 적었지만, 약 65%에 달하는 유사 파일이 존재해 공유 기회를 충분히 활용 할 수 있을 것으로 나타났다. 서로 다른 리눅스 배포판의 경우 윈도우 보다는 낮은 약 35% 비율의 파일 만이 공유 가능성이 있는 것으로 나타났지만 같은 배포판의 경우 버전에 상관없이 거의 100%에 가까운 시스템 파일 유사도를 나타내는 것을 확인하였다. 본 실험을 통해 이러한 디렉토리를 선정해 해당 디렉토리의 파일정보를 전달한다면 가상머신 간의 공유 기회를 대폭 향상시킬 수 있음을 확인하였다.

#### 5. 결론 및 향후연구

본 논문에서 제안하는 공유 시스템은 추가적인 모듈을 추가하여 정보제약을 완화함으로써 불필요한 메모리 비용을 대폭 줄일 수 있으며, 서로 다른 디스크 이미지를 사용하는 게스트 간에도 사용가능하다. 실험을 통해 특정 시스템 디렉토리의 파일 유사도가 매우 높음을 확인하여 게스트가 직접 이러한 정보를 호스트에 전달하는 것이 시스템 전체의 공유 기회를 향상시킬 수 있음을 알 수 있었다. 향후 모니터링 시스템을 실제 구현하여 게스트의 성능 영향 및 실제 공유율의 향상치를 측정하여 본 시스템의 실질적 효용성을 평가 할 것이다.

#### 참고문헌

- [1] C. R. Chang, J. J. Wu, and P. Liu, "An empirical study on memory sharing of virtual machines for server consolidation," *Proc. of the IEEE Int. Symp. on Parallel and Distributed Processing with Applications (ISPA)*, pp. 244-249, 2011.
- [2] C. A. Waldspurger, "Memory resource management in VMware ESX server," *Proc. of the ACM SIGOPS Operating Systems Review*, Vol. 36, No. SI, pp. 181-194, 2002.
- [3] W. Zhao, Z. Wang, and Y. Luo, "Dynamic memory balancing for virtual machines," *Proc. of the ACM SIGOPS Operating Systems Review*, Vol. 43, No. 3, pp. 37-47, 2009.
- [4] A. Arcangeli, I. Eidus, and C. Wright, "Increasing memory density by using KSM," *Proc. of the Ottawa Linux Symposium (OLS)*, pp. 19-28, 2009.
- [5] G. Miłos, D. G. Murray, S. Hand, and M. A. Fetterman, "Satori: Enlightened page sharing," *Proc. of the USENIX Annual Technical Conference (ATC)*, pp. 1-1, 2008.
- [6] K. Miller, F. Franz, M. Rittinghaus, M. Hillenbrand, and F. Bellosa, "XLH: More Effective Memory Deduplication Scanners Through Cross-layer Hints," *Proc. of the USENIX Annual Technical Conference (ATC)*, pp. 279-290, 2013.