ISSN 1226-9182

정보처리학회지

Korea Information Processing Society Review

www.kips.or.kr

KIPS

2016년 7월 | 제23권 제4호 |

스마트 저장 기술

사단법인 한국정보처리학회
KIPS Korea Information Processing Society

글로벌 엔지니어의 길
공학교육인증제도가 있습니다

공학교육인증제도란?

- 공과대학교육과정에 대한 인증을 통해 해당 과정을 이수한 졸업생이 산업체의 요구와 Global Standard를 만족하는 역량을 갖춘 우수인재임을 보증하는 제도입니다.
- 인증졸업생은 Washington Accord회원국(17개국)졸업생과 그 학력의 동등성을 보장 받습니다.

글로벌 공학인재 양성을 선도하는 - 한국공학교육인증원

한국공학교육인증원은 2007년 워싱턴어코드 정회원으로 가입하고, 2008년에는 우리나라 주도로 서울어코드를 창립했습니다. (공학교육인증 졸업생은 회원국 졸업생과 동등한 자격 인정)
2013년, 시드니어코드 및 더블린어코드 정회원이 되었으며 2014년 국내 85개 대학 562개 프로그램이 인증 받았습니다. ▶자세한 사항은 홈페이지 www.abeek.or.kr참조

워싱턴어코드 4년제 공과대학 졸업자 학력의 상호인정을 목표로 설립된 회원국 인증기구간 다자간 국제협의체로 인증기구가 인증한 졸업생은 모든 정회원국에서 학력의 동등성을 보장함.

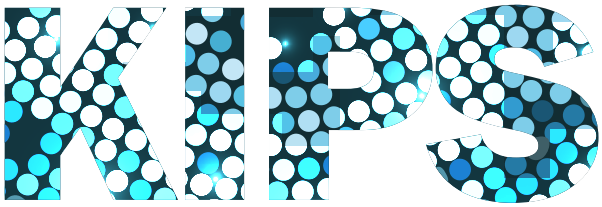
ABEEK 한국공학교육인증원
Accreditation Board for Engineering Education of Korea

▶문의사항 한국공학교육인증원 02-6261-3004

정보처리학회지

Korea Information Processing Society Review

www.kips.or.kr



제 21대 임원명단

김성환 (서울시립대학교)
김우성 (호서대학교)
김태근 (세종대학교)
노웅기 (가천대학교)
박능수 (건국대학교)
손태식 (아주대학교)
안상현 (서울시립대학교)
유진호 (상명대학교)
이경현 (부경대학교)
이의신 (충북대학교)
이필우 (KISTI)
전광길 (인천대학교)
정승원 (동국대학교)
정화영 (경희대학교)
최 욱 (인천대학교)
허의남 (경희대학교)

김수균 (배재대학교)
김인철 (경기대학교)
김평중 (충북도립대학)
노원우 (연세대학교)
박정민 (KIST)
송왕철 (제주대학교)
오세창 (세종사이버대학교)
유환조 (포항공과대학교)
이기훈 (광운대학교)
이재광 (한남대학교)
이화민 (순천향대학교)
전유부 (순천향대학교)
정원용 (원광대학교)
조태남 (우석대학교)
최은미 (국민대학교)
허준범 (고려대학교)

김영욱 (KETI)
김종국 (고려대학교)
김학만 (인천대학교)
문유진 (한국외국어대학교)
박찬열 (KISTI)
신동일 (세종대학교)
우종정 (성신여자대학교)
이영구 (경희대학교)
이원규 (고려대학교)
이재두 (NIA)
임동혁 (호서대학교)
정수환 (숭실대학교)
정윤호 (한국항공대학교)
최강선 (한국기술교육대학교)
추현승 (성균관대학교)

김 용 (한국방송통신대학교)
김종찬 (국민대학교)
김호원 (부산대학교)
민 홍 (호서대학교)
성연식 (계명대학교)
아지즈 (충북대학교)
유윤섭 (한경대학교)
이경오 (선문대학교)
이은영 (동덕여자대학교)
이재호 (서원대학교)
장종수 (ETRI)
정순영 (고려대학교)
정창성 (고려대학교)
최 성 (남서울대학교)
허 경 (경인교육대학교)

협 동 이 사 |

강동석 (NIA)
구태언 (테크앤로법률사무소)
김기범 (국가보안기술연구소)
김성엽 (㈜블루코어)
박철균 (에코메이텍)
안유환 (㈜네오피엠)
이윤재 (SK텔레콤)
이희승 (㈜티노스)
정연수 (Korea IT Times)
최지윤 (㈜한국IT컨설팅)

강태홍 (코스콤)
곽은식 (㈜경봉)
김기철 (한국정보산업연합회)
김완섭 (㈜넥스첼)
박형우 (KISTI)
오형근 (국가보안기술연구소)
이종근 (㈜DSTi)
임종혁 (에이치투오시스템테크놀로지㈜)
지석구 (NIPA)
한영수 (㈜마크애니)

강홍식 (한국전자정보통신산업진흥회)
권문주 (NIPA)
김상열 (대보정보통신㈜)
김용업 (삼성SDS)
서동혁 (㈜영화조세통람)
윤두식 (㈜지란지교시큐리티)
이종주 (㈜시큐브)
정경균 ((주)씨앤앰)
진성철 (유넷시스템)
황일선 (KISTI)

고범석 (㈜자이네스)
김교은 (㈜베스트케이에스)
김성동 (KETI)
김태섭 (㈜바른전자)
서재철 (KISA)
윤인수 (대우정보시스템㈜)
이 철 (LG CNS)
정성무 (KERIS)
최동근 (롯데카드)

지회

강원지회 **김상춘** (강원대학교)
제주지회 **김형수** (제주한라대학교)
호남지회 **서재현** (목포대학교)
중국지회 **Yude Bi** (Luoyang University of Foreign Languages)

영남지회 **김동휘** (대구대학교)
충청지회 **류근호** (충북대학교)
일본지회 **백영선** (코스모컨설팅)

연구회 위원장

e-Bridge **이정배** (부산외국어대학교)
IT정책 **오길록** (숭실대학교)
소프트웨어공학 **박용범** (단국대학교)
에너지그리드정보처리 **이봉재** (전력연구원)
전산교육 **임관철** (대전보건대학교)
전자정부 **이재두** (NIA)
지식 및 데이터공학 **진병운** (ETRI)

IT융합서비스 **박석천** (가천대학교)
빅데이터컴퓨팅 **이필규** (인하대학교)
스토리지시스템 **신범주** (부산대학교)
우정기술 **정 훈** (ETRI)
전산수학 **박진홍** (선문대학교)
정보통신응용 **오진태** (ETRI)
컴퓨터소프트웨어 **박두순** (순천향대학교)

IT시니어봉사단

단 장 | 유기홍 (명지전문대학)

위 원 | 김홍진 (가천대학교) 이준상 (한국IT전문가협회) 정상근 (연성대학교) 정진욱 (인터넷윤리실천협의회)

IT장학사업본부

본 부 장 | 이상범 (단국대학교)

부 본 부 장 | 박정호 (선문대학교)

IT평가인증본부

본 부 장	김병기 (전남대학교)			
부 본 부 장	이상범 (단국대학교)	이영천 (호남대학교)		
위 원	김우성 (호서대학교)	김응수 (대전대학교)	김점구 (남서울대학교)	박석천 (가천대학교)
	박정호 (선문대학교)	박진양 (인하공업전문대학)	박태홍 (LG전자)	윤용익 (숙명여자대학교)
	이병수 (인천대학교)	이임영 (순천향대학교)	조동섭 (이화여자대학교)	조성갑 (前 인천정보산업진흥원)
	최상록 (생산성본부)	최재혁 (신라대학교)	허문행 (안양대학교)	

인터넷윤리진흥본부

본 부 장	정진욱 (인터넷윤리실천협의회)
부 본 부 장	박정호 (선문대학교)

한민족IT평화봉사단

위 원 장	최 성 (남서울대학교)

인사위원회

위 원 장	구원모 (전자신문)			
부 위 원 장	정영식 (동국대학교)			
위 원	유헌창 (고려대학교)	김상훈 (한경대학교)	최유주 (서울미디어대학원대학교)	원유재 (충남대학교)
	조경은 (동국대학교)	박종혁 (서울과학기술대학교)	이장호 (홍익대학교)	이강만 (강릉원주대학교)
간 사	한연희 (한국기술교육대학교)			

포상위원회

위 원 장	김상훈 (한경대학교)			
위 원	최유주 (서울미디어대학원대학교)	원유재 (충남대학교)	조경은 (동국대학교)	박종혁 (서울과학기술대학교)
	문남미 (호서대학교)	한연희 (한국기술교육대학교)	이장호 (홍익대학교)	이근호 (백석대학교)

전임회장 운영위원회

위 원 장	성기중 (프리CEO)			
위 원	조이남 (엑스게이트)	오길록 (숭실대학교)	정진욱 (인터넷윤리실천협의회)	오해석 (가천대학교)
	김홍기 (KTds)	이상범 (단국대학교)	변재일 (국회의원)	김병기 (전남대학교)
	최헌규 (前 다우기술)	이정배 (부산외국어대학교)	금기현 (청년기업가정신재단)	정태명 (성균관대학교)
	오경수 (前 롯데정보통신)	박석천 (가천대학교)	조성갑 (前 인천정보산업진흥원)	박두순 (순천향대학교)

여성위원회

위 원 장	문남미 (호서대학교)			
위 원	길아라 (숭실대학교)	김경아 (명지전문대)	김미혜 (충북대학교)	김미희 (한경대학교)
	박정민 (KIST)	성해경 (한양여자대학교)	송은하 (원광대학교)	신은경 (날리지큐브)
	안상현 (서울시립대학교)	안은영 (한밭대학교)	오수현 (호서대학교)	윤회진 (협성대학교)
	이유부 (성균관대학교)	이은영 (동덕여자대학교)	이정원 (아주대학교)	이화민 (순천향대학교)
	임지영 (성서대학교)	조경은 (동국대학교)	최미정 (강원대학교)	최수미 (세종대학교)
	최유주 (한독미디어대학원대학교)	최은미 (국민대학교)	한영신 (성결대학교)	한정란 (협성대학교)
	홍헬렌 (서울여자대학교)			

학회지편집위원회

JIPS 편집위원회

HwaMin Lee (Soonchunhyang University, Korea)
Hyeoncheol Kim (Korea University, Korea)
Imad Saleh (University of Paris 8, France)
Irfan Awan (University of Bradford, UK)
Jaehwa Chung (Korea National Open University, Korea)
Jangho Lee (Hongik University, Korea)
Jeong-Hyon Hwang (State University of NewYork at Albany, USA)
Ji young Lim (Korean Bible University, Korea)
Jin Gon Shon (Korea National Open University, Korea)
Jin-Hee Cho (U.S. Army Research Laboratory, USA)
Jinsul Kim (Chonnam National University, Korea)
Jongmoo Choi (Dankook University, Korea)
Jongsung Kim (Kyungnam University, Korea)
JooSang Youn (DongEui University, Korea)
JungMin Kim (DaeJin University, Korea)
Jung-Won Lee (Ajou University, Korea)
Kenji Hirata (Toyo University, Japan)
Ki Seok Bang (Hallym University, Korea)
Kibum Kim (Applied Research Center, Motorola, USA)
Ki-Sik Kong (Namseoul University, Korea)
Kuan-Ching Li (Providence University, Taiwan)
Kwang Sik Chung (Korea National Open University, Korea)
Kwangmoon Cho (Mokpo University, Korea)
Kyungbaek Kim (Chonnam National University, Korea)
KyungOh Lee (Sunmmon University, Korea)
Marc Lacoste (France Télécom Division R&D, France)
Mihui Kim (Hankyong National University, Korea)
Milan Markovic (Banca Intesa ad Beograd, Serbia)
Ming Li (California State University, Fresno, USA)
Mohamed Ally (Athabasca University, Canada)
Nam-Mee Moon (Hoseo University, Korea)
Niki Pissinou (Florida International University, USA)
Ouk Choi (Incheon National University, Korea)
Pinaki Ghosh (Atmiya Institute of Technology & Science, India)
Q. Shi (Liverpool John Moores University, UK)
Sanghoon Kim (Hankyong National University, Korea)
Sankar Kumar Pal (Indian Statistical Institute, India)
Seong-Moo Yoo (University of Alabama, USA)
Seungmin Rho (Sungkyul University, Korea)
Shu-Ching Chen (Florida International University, USA)
Soo-Kyun Kim (PaiChai University, Korea)
SoonYoung Jung (Korea University, Korea)
Sung Woo Chung (Korea University, Korea)
Sungsuk Kim (SeoKyeong University, Korea)
Taegeun Kim (Sejong University, Korea)
Taeweon Suh (Korea University, Korea)
Toshiyuki Kamada (Aichi University of Education, Japan)
Wen-Chi Hou (Southern Illinois University, USA)
WonGyu Lee (Korea University, Korea)
WOONG-KEE LOH (Gachon University, Korea)
Yongik Yoon (Sookmyung Women's University, Korea)
Yoon Sok Park (Samsung Electro-Mechanics, korea)
Young-Ho Park (Sookmyung Women's University, Korea)
Yunho Jung (Korea Aerospace University, Korea)
Zhiwen Yu (Northwestern Polytechnical University, China)

Hwayoung Jeong (Kyunghee University, Korea)
Ibrahim Kamel (University of Sharjah, UAE)
Incheon Paik (University of Aizu, Japan)
Jaeho Lee (Seowon University, Korea)
Jaewoo Kang (Korea University, Korea)
Jeonghun Cho (Kyungpook National University, Korea)
Jeong-Joon Lee (Korea Polytechnic University, Korea)
Jiann-Liang Chen (National Taiwan University of Science & technology, Taiwan)
Jin Kwak (Ajou University, Korea)
Jinho Yoo (Sangmyung University, Korea)
Jongeun Lee (Ulsan National Institute of Science and Technology, Korea)
Jong-Myon Kim (University of Ulsan, Korea)
Joongheon Kim (Intel Corporation)
Jun beom Hur (Chung-ang University, korea)
Jungho Kang (Soongsil University, Korea)
Kang-Sun Choi (Korea University of Techonology and Education, Korea)
Keun-Ho Lee (BaekSeok University, Korea)
Ki Yong Lee (Sookmyung Women's University, Korea)
Ki-hoon Lee (Kwangwoon University, Korea)
Kiyoshi Nakabayashi (The Open University of Japan / Chiba Institute of Technology)
Kuniaki Uehara (Kobe University, Japan)
kwangjin Park (Wonkwang University, Korea)
Kyong-Ho Lee (Yonsei University, Korea)
Kyungeun Cho (Dongguk University, Korea)
Lam-for Kwok (City University of Hong Kong, Hong Kong)
Mei-Ling Shyu (University of Miami, USA)
Mi-Jung Choi (Kangwon National University, Korea)
Min Chen (Seoul National University, Korea)
Min-Hyung Choi (University of Colorado at Denver, USA)
Mounir Mokhtari (INT/GET, France)
Neungsoo Park (Konkuk University, Korea)
Omaima Bamasak (King Abdulaziz University, Saudi Arabia)
Pei-Jung Chung (University of Edinburgh, UK)
Ping-Feng Pai (Nation Chi Nan University, Taiwan)
Samadhiya Durgesh (Chung Hua University, Taiwan)
Sang-Soo Yeo (Mokwon University, Korea)
Seng W. Loke (La Trobe University, Australia)
Seung-Ho Lim (Hankuk University of Foreign Studies, Korea)
Shin Byeong Seok (Inha University, Korea)
Simon Fong (University of Macau, Macau)
Soon Ae Chun (City University of New York, USA)
Stefanos Gritzalis (University of the Aegean, Greece)
Sung-Ki Kim (SunMoon University, Korea)
Susumu Kanemune (Osaka Electro-Communication University, Japan)
Taeshik Shon (Ajou University, Korea)
Toshihiro Yamauchi (Okayama University, Japan)
Wanquan Liu (Curtin University, Australia)
Won Woo Ro (Yonsei University, Korea)
Wonyong Jeong (Wonkwang University,Korea)
Yong Kim (Korea National Open University, Korea)
Yoo-Joo Choi (Korean German Institute of Technology, Korea)
Young Hee Kim (Korea Copyright Commssion, Korea)
Yunheung Paek (Seoul National University, Korea)
Yunsick Sung (Keimyung University, Korea)

컴퓨터 및 통신 시스템(KTCCS) 논문지 편집위원회

위 원 장 | 신창선 (순천대학교)

부 위 원 장 | 강승석 (서울여자대학교), 김용석 (서남대학교), 이덕규 (서원대학교), 정광식 (한국방송통신대학교), 최종명 (목포대학교)

위 원 | 강윤희 (백석대학교), 공기식 (남서울대학교), 김승주 (고려대학교), 박광진 (원광대학교), 박능수 (건국대학교), 박재성 (수원대학교), 박희완 (한라대학교), 백상헌 (고려대학교), 윤종희 (영남대학교), 이대원 (서경대학교), 이장호 (홍익대학교), 이태규 (원광대학교), 이화민 (순천향대학교), 이훈재 (동서대학교), 조정호 (광주대학교), 최성곤 (충북대학교), 한영선 (경일대학교), 허 경 (경인교육대학교), 호준원 (서울여자대학교)

소프트웨어 및 데이터 공학(KTSDE) 논문지 편집위원회

위 원 장 | 이은서 (안동대학교)

부 위 원 장 | 박용범 (단국대학교), 이공주 (충남대학교), 전재욱 (성균관대학교), 정재화 (한국방송통신대학교), 조용윤 (순천대학교)

위 원 | 김상근 (성결대학교), 김영철 (홍익대학교), 김우열 (대구교육대학교), 김익수 (숭실대학교), 김인택 (명지대학교), 김정아 (가톨릭관동대학교), 김종호 (순천대학교), 박기남 (고려대학교), 박상준 (군산대학교), 박상현 (연세대학교), 오효정 (전북대학교), 유지환 (한국기술교육대학교), 이대원 (서경대학교), 이상곤 (전주대학교), 이성욱 (한국교통대학교), 이준호 (성균관대학교), 이현아 (금오공과대학교), 정영애 (선문대학교), 최종선 (숭실대학교), 한경호 (단국대학교)

2016년 7월호 특집 담당위원

특 집 위 원 | 임승호 (한국외국어대학교)

공 동 위 원 | 강경태 (한양대학교), 조두산 (순천대학교), 최 민 (충북대학교)

ISSN 1226-9182

정보처리학회지 KIPS

Korea Information Processing Society Review

www.kips.or.kr

2016년 7월 | 제23권 제4호 |

KIPS 권두언

“스마트 저장 기술” 특집을 발간하며...

저장 기술(Storage Technology)이라 함은 PC와 같은 단일 컴퓨터 시스템에서부터 대용량 인터넷 서버 시스템에 이르기까지 컴퓨터 시스템에서 필수적으로 사용되는 저장 관련 하드웨어 및 소프트웨어 기술이라 할 수 있습니다. 컴퓨터 저장 시스템이 사용되는 다양한 방식으로는 DAS(Direct Attached Storage), NAS(Network Attached Storage), SAN(Storage Area Network)에서부터 빅데이터 분산 저장 시스템과 같은 다양한 포트폴리오의 저장 시스템 기술 분야가 있으며, Tape에서부터 HDD(Hard Disk Drive), SSD(Solid State Disks), NVRAM (Non Volatile Random Access Memory)과 같은 매체와 관련된 하드웨어 및 소프트웨어 기술 분야가 있습니다. 최근 정보의 급격한 생성과 저장 매체 기술의 급격한 발전으로 인해서 저장 시스템의 기술적, 산업적 중요성은 더욱 더 증대되고 있다고 할 수 있습니다.

본 특집에서는 스마트 저장 기술을 주제로 선택하면서, NAND 플래시 메모리 및 NVRAM과 관련된 저장 기술 연구 동향에서부터 빅데이터 시스템의 저장 기술 및 클라우드 스토리지 보안 시스템, 저장 시스템의 전력관리 방법 등 다양한 스마트 저장 기술 관련 주제를 포함하고 있습니다. 구체적으로 본 특집의 주제는 NAND 플래시 메모리 기술 동향, NVM/DRAM 하이브리드 메모리 기반의 파일 시스템 연구 동향, 빅데이터 저장시스템인 Key-Value Store의 저장 기술 연구 동향, 클라우드 스토리지 시스템의 보안 관련 기술 연구 동향, 스토리지 서버 시스템의 저전력 관리 기술 연구 동향 등 다섯 가지 주제로 구성되어 있습니다. 이러한 다양한 스마트 저장 관련 기술 동향과 연구 동향을 통해서 점점 증대되고 있는 정보의 저장 및 검색과 데이터 관리 등에 대한 기술 동향과 향후 전망에 대해서 파악하고 더 나아가서 관련 기술을 잘 활용할 수 있으리라 기대합니다.

본 특집호를 위해 원고 집필을 수락해 주시고 원고를 작성해주신 모든 저자 분들께 감사의 말씀을 드리며, 함께 참여해주신 위원장님을 비롯한 편집위원님들과 한국정보처리학회에 진심으로 감사의 인사를 드립니다.

2016년 7월

한국외국어대학교 교수 임 승 호

특집 01

NAND Flash 동작 및 Erase 특성

서주완 (SK하이닉스반도체), 최 민 (충북대학교)

목 차

1. NAND Flash 기술 및 동작소개

NAND FLASH memory는 DRAM(Dynamic Random Access Memory)과 더불어 현재 대표적인 메모리의 형태이나, DRAM과 달리 전원(power)이 공급되지 않는 상태에서도 데이터를 보존할 수 있는 대표적인 비휘발성(Non-volatile) 메모리이다. 특히 NAND형 FLASH memory는 NOR형 대비 cell의 집적도를 높이기가 용이해, 소형화에 유리하여, eMMC(mobile 기기) 및 SSD (Solid State Drive) 등 기기에 널리 이용되고 있다.

NAND FLASH 관련 연구 분야로는 cell의 집적도를 높이기 위한 process integration, 수명을 늘리기 위한 각종 공정 process 개선 및 동작 algorithm 분야, NAND device를 제어하기 위한 controller 분야 등에서 활발한 연구가 진행되고 있다. 앞서 언급한 바와 같이, NAND FLASH는 비휘발성 memory이며 device scaling에 따른 cell 집적화에서도 매우 유리한 장점이 있는 반면에 몇 가지 단점에 분명히 존재한다. 근본적으로 '1'과 '0'의 상태를 만들어주는 동작 mechanism 자체가 floating gate나 well에 high bias(최소 ~10V이상)를 인가하여 charge를 주입하고(program동작), 빼내고 하는(erase동작) FN-tunneling 기반으로 일어나고, 이는 cell의 gate oxide (tunnel oxide) 열화를 필연적으로 수반하기 때문에 결과적으로 지우고, 쓰는 동작 횟수의 제한을 동반하고, 데이터를 보존하는 기간의 제한을 동반한다.

기본적으로 NAND flash에서 program cell(0)과 erase cell(1)을 만드는 방법은 cell gate나 well에 bias를 인가하여 floating gate의 charge potential을 조절하여 cell의 threshold voltage (Vth:문턱전압) 를 원하는 level로 변경시키는 동작이다.

이후, 원하는 level로 만들어진 상태의 cell을 특정 bias read bias를 인가하여 data를 읽어내는(read동작) 동작을 하게 되는데, 이 과정에서 해당 cell의 Vth가 원하는 level대로 위치를 한다면

〈표 1〉 각 data state별 Cell Vth

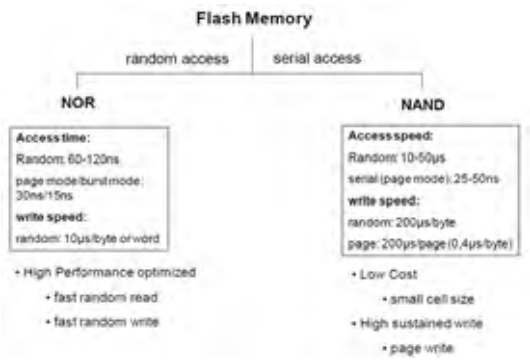
SLC :	Erase cell (1) : Vth → −2V, Program Cell(0) : Vth → 2Vc
MLC :	Erase cell(11) : Vth → −2V, program cell('10') : Vth → 1V, Program cell('00') : Vth → 2V, program cell('01') : Vth → 3V

〈표 2〉 각 data state별 Cell Vth

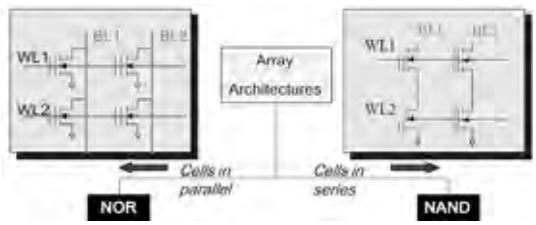
Data(MLC)	11	10	00	01
Cell Vth	−2V	1V	2V	3V

문제가 없으나, 원하는 위치대비 변동이 되어 있다면 잘못된 data를 읽어내게 된다. 따라서, 쓰기 동작(program)이나 지우기(erase) 동작에서 어느 정도의 오차나 산포를 가지긴 하겠지만, target으로 하는 Vth에서 최대한 균일한 Vth를 만들고 유지하는 것이 매우 중요한 과정이라 할 수 있다. 그러나 현실적으로 각각의 cell Vth 상태들은 자기의 Vth를 유지하지 못하고, 여러가지 요인에 의해 변하게 된다. 어떤 요인에 의해 변하게 되는지 각각 살펴보면, program 과정에서 under program, 각 cell의 x, y 방향에서 발생하는 cell 간 간섭(interference), ISPP step 대비 abnormal하게 증가하는 program cell, back pattern에 의한 Vth 변동 등을 가지게 되어, Vth가 원하는 level대비 틀어지게 되어 data를 읽어올 때 잘못 판별된 data를 가져오게 된다. 마이너스(-) Vth를 유지해야 되는 erase cell 관점에서는 program disturbance 현상으로 인해 cell Vth가 +방향으로 Vth가 변하게 되는 현상이 있다. 0V 이상으로 상승시, read 동작시, read level이 0V라면 fail을 유발하게 된다.

이러한 cell간 interference나 disturbance 현상들은 program이나 erase 동작이 반복(EW cycle) 될수록 더 심해지는 특징이 있다. 이는 반복되는 high bias 인가상태에서 벌어지는 FN tunneling 현상으로 인한 tunnel oxide 막질손상(trap site 증가)에 기인한다고 알려져 있다. 이러한 현상들은 구조적으로 오는 피할 수 없는 문제이며, 궁극적으로는 반복적인 EW cycle 기인한 cell의 열화가 있다 하더라도 disturbance나 interference를 최소화 시키는 방법을 찾아 낸다든가, 혹은



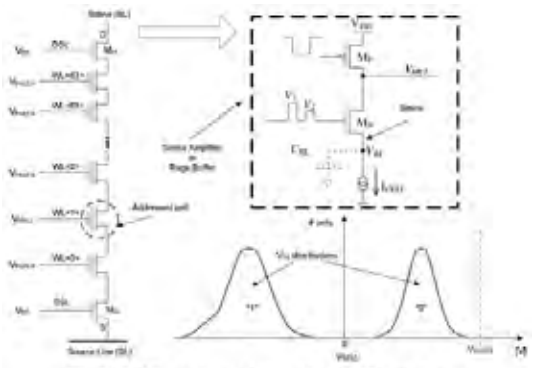
(그림 1) NAND형과 NOR형 FLASH memory의 구분 및 특징



(그림 2) NAND형과 NOR형 FLASH memory의 단위 cell 구조

EW cycle시 stress의 절대량 자체를 줄일 수 있는 방법을 찾아내는 것이 NAND flash 수명을 향상을 위한 중요한 접근 방향이다. 최근 NAND FLASH 수명향상에 관한 방법이 활발하게 연구되고 있는 관계로, 본 고에서는 NAND cell의 기본 동작 원리 및 erase 방식에 대해 소개하고, 각 기존 방식의 특징과 문제점에 대해 설명한다.

NAND flash가 동작속도는 다소 느린 반면, 집적도 면에서는 유리하다.



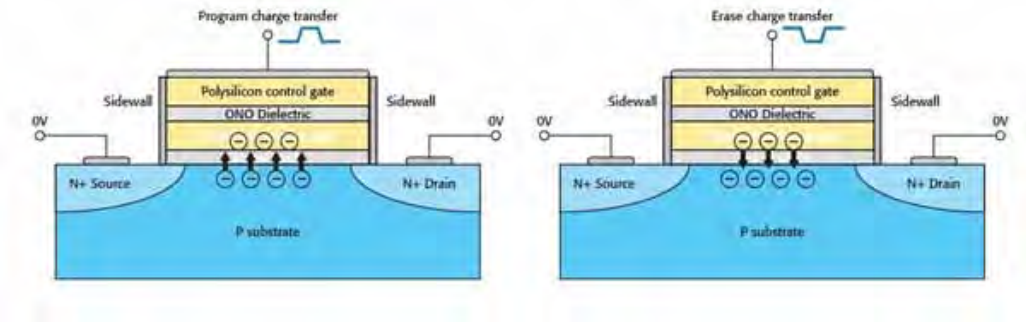
(그림 3) NAND FLASH의 logic '1', '0'의 구분 방법 (Cell Vth)

그림에서와 같이 각각의 cell들은 '0'인 상태를 만들기 위해 program 동작을 해서 양의 값을 가지는 vth를 만들고, '1'의 상태를 만들기 위해 erase 동작을 해서 음의 값을 가지는 vth 를 만들게 된다. 이후, read 동작을 통해, 해당 cell이 '1'인지 '0'인지 판별하게 된다.

좌측이 program 동작으로 bulk에서 floating gate로 전자가 주입되는 과정이며, 우측이 erase 동작으로 floating gate에서 bulk로 전하가 주입되는 동작이며, 둘 다 FN-tunneling 동작으로 charge가 이동한다.

Program verify와 Erase verify 동작에 대해 간략히 알아보면, program이나, erase 상태의 cell vth를 형성을 한 후, 추후 read시 '1', 또는 '0'의 판별을 하기 위해서는, 임의의 voltage level이 아닌 원하는 target level을 설정해 놓게 되는데, 이 level을 program verify level 혹은 erase verify level이라 하며, 그 지점에 도달 할 때 까지 program 이나 erase bias를 인가하고, verify level이 넘어가는 cell들은 inhibit 동작(ex. PGM cell의 경우 bitline에 Vcc인가하여, FN tunneling 동작 비활성화)을 하여, 더 이상 Vth 변화가 일어나지 않도록 한다.

Process가 진행되고 난 뒤, 동작을 거의 하지 않은 초기 상태의 NAND flash cell이라면, 위에서 기술한대로 cell vth를 원하는 level로 문제 없이 가져다 놓을 수 있고, 그렇게 변화된 Vth를 읽어내어 '1', '0'을 판별해 내는 것도 대체로 문제가 없다. 정작 문제는 data를 썼다, 지웠다하는



(그림 4) 단위 cell transistor 에서의 program 및 erase 시, bias 조건 및 전하의 움직임.

동작을 반복하게 되면(Program, Erase동작 반복-EW cycle) 초기 상태의 cell에 맞는 Vth를 맞추기 위한 bias로 동작을 시키게 되면 Vth 변화가 생기게 되는데, 일반적으로 program 동작시에는 cell이 더 빨리 vth level에 도달하는 동시에 (speed가 빨라짐), Vth 산포가 열화되는 특징이 있으며, erase 동작시에는 cell이 더 늦게 Vth level에 도달하게 되는 동시에(speed가 느려짐), 역시 Vth 산포가 열화되는 특징이 있다. 따라서, 일반적으로 EW cycle후에 program 동작시에는 더 낮은 program bias가 필요하고, erase 동작시에는 더 높은 erase bias를 인가해야지만, 안정된 Vth 산포를 가지면서 원하는 verify level로 위치시킬수 있다.

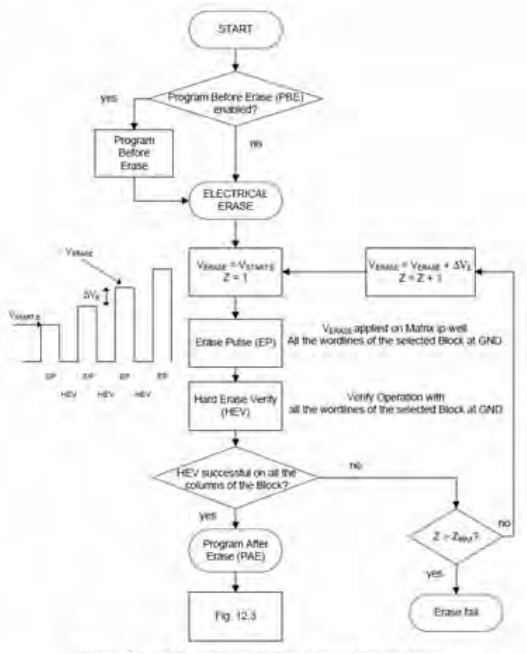
앞서 기술 한 바와 같이, program이나, erase시 bias를 높게 사용하면, ISPP나 ISPE동작시에 그 만큼 빠른 시간에 원하는 cell Vth level로 만들 수 있기 때문에 유리한 면이 분명히 있지만, (지우기, 쓰기 속도 향상) Gate oxide로 전자가 tunneling 하는 구조다보니, 특히 erase동작의 경우, 동작 bias가 높을수록 막질 열화가 심해지고, Vth shift 현상 및 Vth distribution 열화가 더 심해지고, 그렇게 되면 bias를 더 높게 써야 되고, 결국에는 원하는 수명 연한을 보장 할 수 없게 된다.

따라서, 수명 연한은 최대한 보장하되, program이나 erase 속도는 최대한 빨리하면서, Cell Vth distribution 열화를 최소화 하는 동작 bias를 결정하는 것이 현재까지도 매우 중요한 과제이다.

2. NAND Flash Erase 특성소개

NAND Flash 메모리에서 erase 동작시 erase pulse를 인가한 후, erase Vth가 원하는 level이하로 제대로 도달했는지 여부를 확인하기 위해, erase verify 동작을 수행한다. 기본적으로 verify 동작은 (program verify 동일) ‘read’ 동작과 유사한 동작이다. Program이나 erase 동작이 끝난 후, ‘read’ 하는 동작은 해당 cell이 최종적으로 ‘1’ 인지 ‘0’인지 판단 하는게 목적이라면 erase나 program 중간 중간에 들어가는 verify 동작은 원하는 Vth까지 도달했는지 확인하는 목적의 동작이다. 따라서 erase 동작의 경우, erase pulse를 인가하는 과정상의 동작이므로, verify 동작이나 조건이 후속 erase bias를 결정짓는데 직접적으로 영향을 준다.

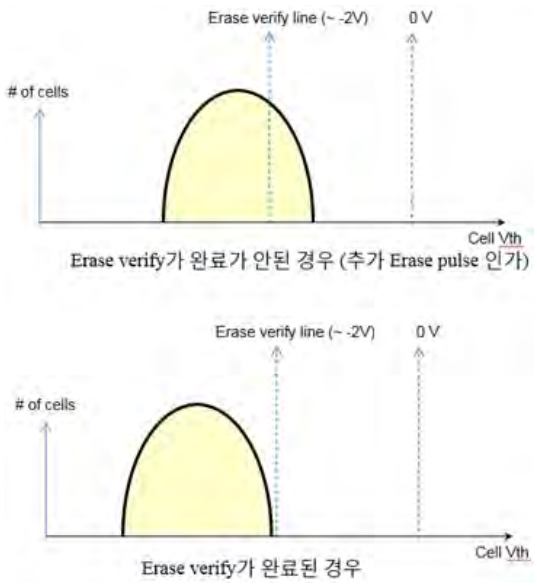
기존 erase verify의 경우, verify 동작시 음(-)의 level에 모든 cell들이 제대로 도달했는지 확인하기 위해, 모든 WL에 동일한 bias를 인가하여 cell들이 erase 되었는지 확인을 한다. (일반적인 ‘read’ 동작시에는



(그림 5) ISPE 알고리즘 flow chart

read 하고자 하는 WL에만 원하는 bias(0V~5V 정도) bias를 인가하고, 나머지 WL에는 pass bias (~6V이상)를 인가하므로 erase verify 동작과는 bias 조건이 약간 다르며, 음(-)의 level의 cell Vth 상태를 판별해 낼 수 없다) 기존 erase verify의 경우, cell string 전체의 WL에 bias를 인가하다보니, cell Vth variation이 있는 경우, cell string에 흐르는 전류(current)가 작게 흐를 수 밖에 없는 동작 조건이고, erase verify가 실패할 확률이 상대적으로 높아지게 된다.

일반적으로 워드라인(wordline)에 0V~1V 정도의 bias를 인가하게 되는데, erase가 충분히 다 잘되어 있는 경우라면 문제가 없겠지만, string내의 특정 한 cell이라도 variation을 가져서 erase speed가 느린 경우라면, 그 cell로 인하여 erase verify에 실패하게 되고, 결국 ISPE 동작 알고리즘상 더 상향된 bias로 다시 한 번 erase 동작을 수행하게 된다. 그림 3은 ISPE 알고리즘의 플로우 차트를 나타낸다.



(그림 6) Erase Verify 동작 개념

만약 erase verify에 실패한 경우라면, 위 그림 6에서 도시된 바와 같이 verify하고자 하는 bias와 원래 verify하고자 하는 bias 전압보다 얼마간의 bias를 더 추가로 인가함으로써, 2회에 걸쳐 verify를 수행하는 방식 등을 택할 수 있다.

3. 결 론

NAND Flash 메모리 기술은 빠른 속도로 발전하고 있으며, 과거 SLC 방식으로부터 MLC, 현재는 TLC 방식이 널리 사용되고 있으며, 최근 삼성에서는 테라바이트급 SSD를 양산하는 추세에 있다. 실제로 NAND Flash 메모리 기술은 제품에서 활용하는데 있어 위에서 설명한 바와 같이 물리적인 측면에서 고려해야 할 부분도 많고, 추가적으로 연구하여야 할 부분도 많다. 또한, NAND Flash 메모리 제품의 용량확대 측면에서 3D Stacking 기술에 대해서도 최근 많은 고려가 있다. 뿐만 아니라, 이러한 NAND Flash 제품에 대한 최종 사용자 관점 인터페이스 및 NAND Flash 메모리 사용/활성화/보급확대 측면에서는 임베디드 소프트웨어 기술이 혁혁한 공헌을 한 바 있다. 앞으로도 국내외 반도체 제조사, 연구소/대학 등 연구기관, 임베디드시스템 전문가 들의 노력에 힘입어 더욱 편리하고 유용한 휴대단말 기기들이 탄생할 것을 기대하면서 본 고를 마칩니다.

참 고 문 헌

[1] Inside nand flash memories, Rino Micheloni, Springer

저 자 약 력

서 주 완

이메일 : cantab@naver.com

- 2004년 광운대학교 전자재료공학과 (학사)
- 2016년 충북대학교 정보통신공학부 (석사)
- 2004년~현재 SK하이닉스반도체 책임연구원
- 관심분야 : NAND Flash 메모리 공정, 임베디드 시스템

최 민

이메일 : mchoi@cbnu.ac.kr

- 2009년 한국과학기술원 전산학과 (박사)
- 2008년~2010년 (주)삼성전자 책임연구원
- 2011년~현재 충북대학교 정보통신공학부 교수
- 관심분야 컴퓨터 시스템, 임베디드 시스템

특집 02

NVM/DRAM 기반 하이브리드 메인 메모리를 위한 파일 시스템 동향

조중석 · 조두산 (국립순천대학교)

목 차

1. 서 론

DRAM은 실제 프로세스들이 작업을 수행하는 메인 메모리 장치로 디스크 다음으로 넓은 용량이 사용되고 있다. 지금까지 10년에 약 100배의 비율로 30년 이상 그 용량을 확대하여 개발되어왔다. 프로세서 속도 향상을 나타내는 무어의 법칙과 더불어 IT기술 발달을 지탱해온 기둥이었다. 하지만 엘피다메모리사의 CTO가 말했던 것처럼 4~5년전의 DRAM 제조 기술로는 30nm 이후 공정에서는 미세 공정에서의 개발이 매우 어려워진다고 전망되었었다. 콘덴서와 트랜지스터 조합으로 데이터를 기록하는 DRAM은 콘덴서 때문에 미세화가 어렵게 된다. 데이터 저장을 위해서는 전위차를 만들 콘덴서가 필요한데, 30nm 이후 미세공정에서는 최소 용량인 20 femto farad을 확보할 수 없게 되기 때문이다. 귀금속을 재료로 사용하는 몇가지 해결책이 있으나 당시의 메모리 단가에서 수배 가격 상승을 발생시키기 때문에 제품 경쟁력이 떨어진다. 이러한 미세화에 대한 해결책으로 많은 기업에서 비휘발성 메모리 기술을 선택하였다. 왜냐하면 4~5년 전에는 공정 미세화의 한계로 비휘발성 메모리와 같은 차세대 메모리 기술이 기존 DRAM을 대체 할 것이라고 예상했기 때문이다. 그러나 이러한 메모리 기술이 예상을 깨고 14나노까지 상용화되었고, 10나노 이하 기술이 등장할 전망인데다 3차원 반도체 적층 기술과 멀티 패터닝 기술 등 집적도를 높이기 위한 설계 · 공정 · 소재 기술이 발전함에 따라 기존 메모리의 '대체'가 아닌 '보완' 형태가 상용화될 것으로 보인다. 즉, 상호보완적인 하이브리드 메모리의 형태로 개발되는 것이다.

삼성전자와 SK하이닉스는 이미 D램과 낸드플래시 등 기존 메모리의 성능을 높이고 단점을 보완할 수 있는 새로운 메모리 제품군을 개발하는 데 속도를 내고 있다. 새로운 제품을 개발해 시장을 창출하는 형태가 아니라 시장 요구를 맞추기 위한 보완

적인 형태인 것이다. 상용화에 성공하면 새로운 성장동력으로 빠르게 자리매김할 전망이다.

과거와 달라진 것은 차세대 메모리가 이미 시장에서 확고한 위치를 차지한 D램, S램, 낸드플래시를 대체하기보다 이들의 성능을 보완해주는 데 초점이 맞춰있다는 것이다. 현재 가장 주목받는 차세대 메모리 기술은 3D XPoint, R램(저항변화형 메모리), P램(상변화 메모리), STT-M램(스핀주입 자화반전 메모리) 등인데 우리는 여기서 이러한 메모리 기술을 통칭 비휘발성 메인 메모리 (Non-Volatile Main Memories, NVMM)으로 표기하겠다. 각각의 메모리 기술들이 서로 다른 장단점을 갖고 있는데, STT-MRAM은 DRAM보다 빠르기 때문에 온칩 메모리 혹은 하단 캐시 (last-level cache)로 쓰일 수 있으나 큰 셀 크기 때문에 용량에 제약을 받아 DRAM을 대체하기는 어렵다고 전망되고 있다. PCM과 ReRAM은 DRAM에 비하여 집적도가 높으나 접근속도가 느리기 때문에 DRAM을 단일 기술로 대체하기에 용이하지 않다. 3D XPoint 메모리 기술은 최근 인텔과 마이크론사에 의하여 발표되었다. 낸드플래시에 비하여 1000배까지 속도 향상이 가능한 것으로 알려져 있다. 조만간 SSD를 대체하는 기술로 자리매김 할 것으로 전망된다.

우리는 여기서 이러한 NVMM과 DRAM의 단점을 보완하고 상호 장점을 살려주는 하이브리드 메인메모리에 대해서 살펴보기로 한다. 앞서 언급된 바와 같이 NVMM은 미세 공정에서 집적도를 충분히 활용할 수 있도록 하는 메모리 기술이며, 현재 이를 활용한 저전력 연구도 많이 진행되고 있으나 산업계에서는 실질적으로 집적도 미세화에 따른 수율을 맞추기 위한 방안으로 선택하여 활용되고 있다. 하이브리드 메모리가 개발되면 결국 상용화의 성공을 결정할 가장 중요한 요소는 파일 시스템이다. 기존 파일 시스템은 메모리 계층 성능 최적화를 위하여 단일 디스크와 DRAM 사이의 대역폭을 효율적으로 활용하는데 초점을 맞추어 개발되어 있다. 하지만 하이브리드 메모리에서는 대역폭의 활용과 더불어 NVM의 장점을 최대한 활용한 데이터 일관성 (consistency) 유지가 중요한 문제로 대두된다. 우리는 여기서 하이브리드 메모리의 파일 시스템에서 요구되는 주요한 문제들을 살펴보고자 한다. 비휘발성 메모리의 장점 때문에 NVMM 기반의 하이브리드 메인메모리가 곧 시장을 점유하기 시작할 것으로 예상되며, 이때 하이브리드 메모리를 구성하는 각 메모리의 장점을 충분히 활용하도록 설계된 파일 시스템의 특징에 대해서 기존 연구[1]를 기반으로 살펴보고자 한다.

2. DRAM과 NVM으로 구성된 하이브리드 메모리에 적합한 파일 시스템

상대적으로 빠른 휘발성 메모리인 DRAM과 집적도가 높고 약간 상대적으로 느린 상변화메모리 같은 비휘발성 메모리 결합은 상호 기술의 장점을 극대화하는 최적의 방안으로 그 가능성을 주목받고 있다. 하이브리드 메인 메모리 시스템은 매우 강력한 일관성 (consistency) 보장을 위한 부분이 필요하며 동시에 소프트웨어 오버헤드도 최소화해야 하이브리드 메모리의 장점을 충분히 활용할 수 있게 된다. 이러한 관점에서 기존의 파일시스템은 하이브리드 메모리 시스템에 적합하지 않다. 기존 기법은 디스크의 성능에 초점을 맞추어 개발되어 있으며 일관성 또한 디스크의 정확성 (correctness)에 의존하기 때문이다. 강력한 일관성 보장은 특히 NVMM 기반 파일시스템에서 도전적인 요소이다. NVMM에서 쓰기 동작은 큰 비용을 요구하기 때문이다. 현대의 CPU와 메모리 시스템은 성능 개선을 위하여 쓰기 동작 순서를 변경하기도 한다. 이때 시스템 오류가 발생하면 일관성이 유지되지 못한다. 이

러한 경우를 방지하기 위하여 파일시스템은 CPU 캐시에서 데이터를 순서를 명확히 유지하여 플러시 (flush) 되도록 할 필요가 있다. 이때 순서 유지를 위한 오버헤드가 추가되고 NVMM을 통한 성능 개선을 포기하더라도 일관성 유지를 위해 이러한 부분들을 희생해야하는 문제점이 발생한다.

많은 애플리케이션들은 원자적 파일 시스템 동작 (atomic file system operation)에 그들의 정확성을 의존하기 때문에 이러한 문제를 해결하는 것은 중요하다. 현존하는 주요 파일 시스템은 저널링 (journaling), 쉐도우 페이징 (shadow paging), 로그 구조 (log structuring) 기술들을 이용하여 원자성 (atomicity)를 제공한다. 하지만 저널링은 쓰기 연산의 증가에 따라 대역폭을 낭비하게 되고, 쉐도우 페이징은 잎 노드 (leaf node)에서 루트(root)까지 연속적인 업데이트를 요구하기 때문에 양쪽 기술 모두 쓰기 순서 제약 (ordering constraint)을 유지하기 위한 성능 저하를 피할 수 없게 된다.

하드디스크 혹은 NAND 플래시 기반 SSD에서 효과적으로 사용할 수 있도록, 로그 구조화 파일 시스템 (log-structured file system, LFS)은 적은 수의 랜덤 쓰기 요청을 더 많은 순차 쓰기로 그룹화할 수 있다. 하지만 현재의 LFS는 가용한 연속된 자유 공간 (contiguous free region)에 의존적이다. 연속된 자유 공간을 유지하기 위해서는 비싼 비용을 요구하는 가비지 콜렉션 동작 (garbage collection operation)을 빈번히 실행해야 한다. 결과적으로 최근의 연구[2] 에서는 NVMM에서 LFS가 저널링 파일시스템보다 성능이 저하될 수 있음을 보고하였다. 이러한 여러 가지 어려운 문제들을 해결한 결과가 최근 학계에 보고되었다 [1]. 우리는 여기서 해당 케이스를 살펴보고 향후 흐름에 대한 전망을 논해보기로 한다.

2.1 NOn-Volatile memory Accelerated (NOVA) 로그구조 (log-structured) 파일 시스템

NOVA는 여러 부분에서 기존의 로그 구조 파일 시스템과 다르다. NOVA는 각각의 inode를 분리된 로그로 구성하여 보통 동작과 복구 (recovery) 실행동안 동시성 (concurrency)을 최대화 하도록 하였다. NOVA는 로그들을 링크리스트 (linked list)로 저장하기 때문에 연속된 메모리 공간이 필요없다. 로그의 끝 (tail) 포인터에 원자적 정보 갱신을 수행하기 때문에 경량 저널링이 가능하다. NOVA는 데이터 로그를 기록하지 않기 때문에 복구 프로세스는 NVMM의 일부만을 스캔하는 것으로 경량화 된다. 이로써 가비지 콜렉션의 오버헤드를 줄일 수 있고 메모리가 거의 차있는 상황에서도 좋은 성능을 제공할 수 있게 된다.

2.2 배경 지식

NVMM 기술은 파일 시스템 디자이너에게 여러 가지 도전적 과제를 안겨주었다. 가장 중요한 것 중 하나는 소프트웨어 오버헤드에 대한 메모리 성능의 밸런싱이다. 이와 관련된 요소들은 다음과 같다.

- **성능 :** NVMM의 낮은 지연시간 (latency)은 하드웨어와 소프트웨어 지연시간 사이의 상충관계 (trade-off)를 변화시켰다. 보통 저장 시스템은 높은 지연시간을 갖는 디스크와 같은 저장장치가 접근 시간을 결정하는 주요요인이다. 그래서 소프트웨어 효율은 중요한 요소가 아니었다. 하지만 최근 몇몇 연구결과 [3, 4]가 보였듯이 빠른 NVMM에서는 소프트웨어 비용이 접근시간을 결정하는 주

요요인이 된다. STT-MRAM 기반의 NVMM은 낮은 지연시간을 제공하기 때문에 프로세서의 메모리 버스에 직접 연결되고 로드/스토어 (load/store) 명령어로 직접 사용된다. 최근의 NVMM 기반 파일시스템은 DRAM 페이지 캐시를 거치지 않고 직접 접근 기술 (Direct Access, DAX) 혹은 eXecute In Place (XIP)로 불리는 기술로 NVMM과 DRAM 사이의 저장 스택에 추가 복사본을 만들지 않고 NVMM에 직접 접근된다. NOVA의 경우 DAX 파일시스템이고 향후의 NVMM 파일시스템도 유사한 유형으로 개발 및 사용될 것이 명확하다.

- **쓰기 재정렬 (Write reordering):** 현대의 메모리 계층에서 프로세서와 캐시는 쓰기 명령 순서를 재정렬하여 성능 개선을 할 수 있다. CPU의 메모리 일관성 프로토콜은 메모리 쓰기 순서에 대한 보장을 제공한다. 그러나 현재 사용되는 이러한 기술이 NVMM에는 적용되지 않는다. NVMM에서 전력 손실에 의한 일관성 복구 절차는 메모리 쓰기 순서 변경을 완전히 복구할 수 없기 때문에 NVMM에 한해서는 쓰기 순서 변경에 의한 성능 개선 부분은 희생해야한다. 이러한 문제를 해결하기 위한 NVMM 인지 소프트웨어는 명확한 캐시 플러싱 (cache flushing) 명령어를 사용하여 쓰기 순서를 유지할 수 있다. x86 아키텍처는 CPU 캐시라인을 플러시 하도록 clflush 명령어를 제공한다. clflush 명령어는 캐시라인들이 메모리에 올바른 순서에 따라 기록되는 것을 보장하지만 해당 캐시라인들은 무효화되기 때문에 캐시의 효율적인 활용에 따른 장점을 잃게되고 이에 따라 전체적으로 성능이 현저하게 저하된다. 게다가 명확히 말하면 메모리에 기록되는 것이 보장되는 것이 아니라 메모리 제어기 (controller)에 전달되는 것만 보장하기 때문에 데이터가 메모리에 기록되는 것을 완전히 보장하지는 못한다. 이러한 문제를 해결하기 위해서 인텔은 clflush 명령어의 효율화 버전인 clflushopt를 추가하였다. NOVA는 이러한 개선 명령어에 기반하여 구성되었다.
- **원자성 (atomicity):** POSIX 스타일의 파일 시스템 시맨틱 (semantic)은 여러 동작들이 원자적 (atomic)이 되는 것을 요구한다. 즉, 모든 작업이 완전하게 실행되거나 아니면 전혀 실행되지 않거나 양자택일되는 결과가 발생하도록 동작되는 것이다. 예를 들어, POSIX rename 명령은 명령이 실패하였을 때 해당 파일 이름이 원래의 이름으로 남아 있거나 혹은 새로운 이름으로 변경되거나 양자 경우 중에서 어느 한가지 결과로 종료되는 것이 보장되어야 한다. 파일이름 변경은 메타데이터만을 변경하는 작업이지만 다른 원자적 명령은 데이터와 메타데이터 모두를 수정하는 경우도 있다. 때로는 원자성을 제공하기 위하여 프로그래머는 새로운 복잡한 기법을 직접 개발해서 사용해야 한다. 현재 원자성 보장은 기본적인 작업에서만 제공되고 있다. 예를 들면, 디스크에서 정렬된 섹터 쓰기 (aligned sector writes)는 8바이트 이하 단위에서만 제공된다. 하이브리드 메모리에서는 이것을 유지하기 위한 새로운 방안을 모색해야 할 것이다.

2.3 복잡한 원자적 동작 보장

현존하는 파일 시스템들은 저널링, 쉐도우 페이징, 로그 구조화 등 다양한 기술들을 사용하여

원자성 보장을 제공한다. 이와 관련하여 살펴보도록 하자.

- **저널링:** 저널링 시스템은 저널에 모든 변경사항을 실제 변경이 이루어지기 이전에 기록한다. 전력 손실 오류가 발생하는 경우에 저장된 저널에 따라 시스템의 일관성을 유지하도록 복구하는 것이 가능하게 된다. 따라서 저널링은 데이터 쓰기를 저널 로그에 한번 변경 대상에 한번 이렇게 총 두번 수행하게 된다.
- **쉐도우 페이징:** 기존의 다양한 파일 시스템이 쓰기 복사 (copy-on-write) 메커니즘을 사용하고 있다 [5, 6]. 쉐도우 페이징 파일 시스템은 원자성을 제공하는 트리 구조에 강하게 의존한다. 쓰기 시 해당 데이터를 바로 수정하는 대신에 쉐도우 페이징 쓰기는 새로운 복사본에 수정을 수행한다. 새로 생성한 복사본은 저장장치의 빈공간에 생성한다. 이때 새롭게 생성된 페이지들이 파일 시스템 트리를 갱신하게 만드는데, 이 추가된 페이지인 잎노드에서 루트 노드까지 트리 자료구조의 연속된 업데이트 작업은 잠재적인 오버헤드 비용으로 연결된다.
- **로그 구조 파일 시스템:** 로그 구조 파일 시스템 (log-structured file system, LFS)는 본래 하드디스크 드라이브의 순차접근에 의한 성능 개선을 이용하기 위하여 설계되었다. LFS는 메모리의 랜덤 쓰기 연산을 그룹화 (grouping)하고 결과적으로 더 많은 순차쓰기 동작으로 이들을 변환하여 대역폭을 충분히 사용함으로써 하드디스크 성능을 최적화하게 된다. LFS는 명쾌한 아이디어임에 분명하지만 이것을 효과적으로 구현하는 것은 어려운 일이다. 왜냐하면 LFS는 순차쓰기를 디스크의 연속된 공간에 진행해야 하고, 이러한 공간 확보를 위하여 빈번한 가비지 콜렉션을 수행하여야하기 때문이다. 이러한 오버헤드를 줄이기 위한 몇몇 시도들이 현재 학계에 보고되어있다. F2FS [7]는 여러개의 로그 헤드를 사용하여 메타데이터와 데이터 로그를 분리한다. 새로운 데이터는 현재 있는 자유공간 (free space)에 바로 쓰도록하여 가비지 콜렉션 오버헤드를 줄일 수 있었다. RAMCloud [8]는 DRAM 기반 저장장치에서 모든 로그 구조를 기록하고 관리하도록 하여 DRAM 사용율을 높이도록 설계되었다. 또한 백업 데이터는 디스크에 두고 자유공간 확보가 원활하도록 지원하였다.

2.4 NVMM을 위한 파일 시스템

몇몇 연구 그룹에서 NVMM 기반 파일시스템의 문제점들을 고려하고 그에 대한 해결책을 발표하여 왔다. BPFS [9]는 메타데이터와 데이터의 일관성을 제공하는 쉐도우 페이징 파일시스템 이다. BPFS는 쓰기 복제와 순서 일관성을 제공하기 위한 하드웨어 메카니즘을 사용하였다. 하드웨어 쉐도우 페이징은 복사 및 관리에 대한 소프트웨어 오버헤드를 제거할 수 있으나, 고정된 하드웨어 버퍼 크기 때문에 거대한 파일시스템에서는 이러한 오버헤드가 다시 문제로 재발할 수 있는 단점이 존재한다. PMFS [10]는 경량화된 DAX 파일시스템이다. 메타데이터 수정에 대한 저널링을 사용하지만 데이터 바로 쓰기를 허용하여 원자성 제공을 못하고 결과적으로 데이터 일관성이 보장되지 않는다. Ext4-DAX [11]는 NVMM에 직접 접근 가능하도록 Ext4를 DAX에 적합하게 확장하였다. 또한 원자성이 보장된 저널링으로 메타데이터 수정을 지원한다. DAX로 확장되지 않은 본래의 Ext4 파일시스템은 데이터 원자성을 보장하는 데이터 저널 모드를 포함하고 있으나, 확장된 파일시스템은 해당 모드를 지원하지 않아 데이터 수정에 원자성이 제공되지 못하고 있다. SCMFS [12]는 운영체제의

가상 메모리 관리 모듈을 사용한다. 이로써 가상의 연속된 주소 영역으로 파일을 매핑하고 관리 및 접근을 단순화하고 경량화 할 수 있었다. 다만 메타데이터와 데이터의 일관성 보장을 위한 기능이 제공되지 않았다. Aerie [13]은 NVMM 데이터 접근을 위한 기능과 파일 시스템 인터페이스를 구현하였다. POSIX 시맨틱을 완화하여 성능을 개선하였으며 메타데이터 저널링을 제공하였다. 하지만 데이터 원자성과 mmap 동작에 일관성 지원을 하지 못하였다.

3. NOVA 설계안

NOVA는 하이브리드 메모리에서 LFS의 장점을 충분히 활용하도록 POSIX 파일시스템으로 구성되어있다. 두 가지 서로 다른 메모리 기술로 구성된 하이브리드 메모리에는 기존의 디스크 대역폭을 최대로 활용하도록 구성된 기존의 파일 시스템과는 다른 구조의 특징들이 요구된다. NOVA는 3가지 주요 분석결과를 바탕으로 설계되었다. 첫째, NVMM에서 원자적 수정을 지원하는 로그는 구현이 쉽지만 디렉토리 검색과 파일 액세스와 같은 검색 동작은 효율적이지 않다. NVMM에서 빠른 검색을 지원하는 트리와 같은 자료구조는 더욱 정확한 구현이 어렵다 [14, 15]. 둘째, 로그 삭제의 복잡도는 연속된 자유 공간 확보를 지원하는 방식에 따라 결정되는데, NVMM에서는 랜덤 접근 비용이 저렴하기 때문에 복잡한 방식이 요구되지 않는다. 셋째, 단일 디스크에서는 공간 지역성을 개선하기 위하여 단일 로그를 사용하는 것은 효과적인 방법이지만, NVMM에서는 빠른 광대역 동시 접근이 가능하기 때문에 복수의 로그를 사용하는 것이 성능에 긍정적인 효과를 가져올 수 있다.

이러한 3가지 관찰을 바탕으로 NOVA는 다음과 같은 5가지 특징을 갖는 파일 시스템으로 설계되었다.

- NOVA는 로그와 파일 데이터를 NVMM에 저장하며 빠른 검색을 수행하기 위한 radix tree [16]를 DRAM에 저장한다. radix tree는 리눅스 커널에서 다양하게 사용되었고, 많은 테스트를 거쳤으며, 충분히 완성도가 높아 NVMM에서 자료구조를 단순하고 효율적으로 구성하여 준다.
- 각각의 inode가 자신의 로그를 갖도록하여 동기화 없이 동시 수정이 가능하도록 한다. 이렇게 함으로써 NOVA는 복구시 다중 로그를 동시에 재현 (replay) 할 수 있으며 파일 접근 또한 높은 동시성 (concurrency)을 제공할 수 있게 된다.

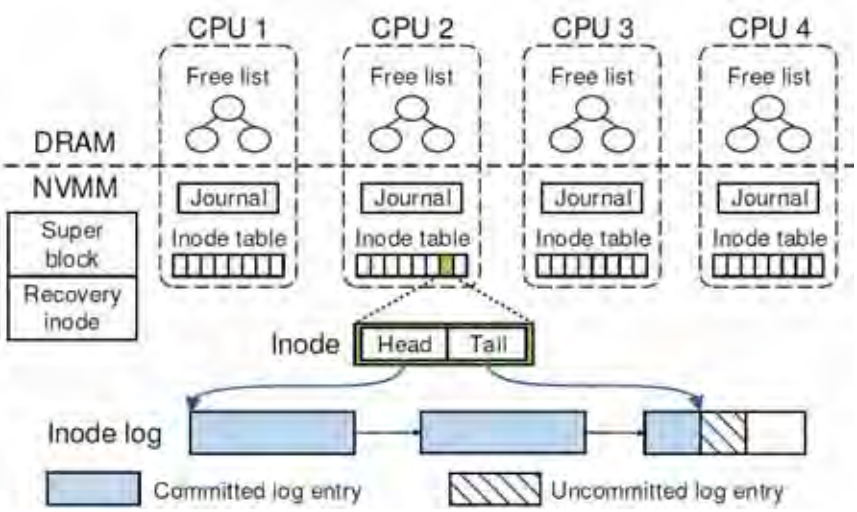
NOVA는 로그 구조 파일 시스템인데 이점은 저널링과 쉐도우 페이징 보다 더 저렴한 비용의 원자적 수정을 제공할 수 있게 한다. 로그에 원자적 데이터 쓰기를 하기 위해서 NOVA는 우선 데이터를 로그에 끼워두고 그 다음에 실제 수정을 실행하기 위해서 로그 끝 (tail)을 수정하게 된다. 이렇게 하면 저널링의 쓰기 중복 오버헤드와 쉐도우 페이징의 연속된 자료구조 갱신 오버헤드를 피할 수 있게 된다. move와 같은 디렉토리 관련 동작 실행시 여러 inode들을 선회하게 되는데, NOVA의 경우 복수의 로그들을 저널링을 통하여 원자적 (atomically) 업데이트를 수행한다. NOVA의 저널링은 데이터를 각 inode의 로그 끝에 쓰기 때문에 오직 로그 끝 (tail)만을 업데이트하면 된다. 이러한 점은 파일 데이터 혹은 메타 데이터와는 다르게 NOVA의 저널링을 상당히 경량화 시켜주는 요인이 된다.

디스크와는 다르게 NVMM에서는 지역성에

기반한 순차 로그의 장점이 상대적으로 작다. 광대역 동시 랜덤 접근이 가능하기 때문이다. 따라서 NOVA는 4KB NVMM 페이지를 위한 하나의 링크리스트에 로그와 다음 페이지를 위한 포인터를 저장한다. 이처럼 비순차적인 로그 구조는 3가지 이점을 제공하는데, 우선 대량의 연속된 공간이 필요없기 때문에 로그 공간 확보가 용이해 진다. 다음으로 로그 삭제가 페이지 크기 단위로 정밀하게 진행될 수 있다. 마지막으로 오래된 엔트리를 포함하는 로그 페이지들의 환원(reclaiming)을 단지 몇개의 포인터 할당만으로 수행할 수 있는 이점이 있다.

NOVA의 inode 로그들은 파일 데이터를 포함하지 않는다. NOVA는 수정된 페이지들을 위한 쓰기 복사 (copy-on-write)를 사용하고 로그에 그 쓰기에 대한 메타데이터를 첨부한다. 이 메타데이터는 해당 데이터 페이지와 업데이트를 기록한다. 파일 데이터의 쓰기 복사는 여러 이유로 상당히 유용하다. 우선 복구 절차를 가속할 수 있는 더 짧은 로그를 구성할 수 있게 하고, 가비지 콜렉션을 더 단순하고 효율적으로 만들어 준다. NOVA는 로그 페이지를 복원할 때 해당 로그로 부터 파일 데이터를 복사하지 않기 때문이다. 다음으로 DRAM 프리리스트 (free list)에서 페이지를 추가 삭제만으로 이전 페이지 복구와 새로운 데이터 페이지 할당이 가능하기 때문에 구현이 용이하다. 마지막으로 데이터 페이지 복구를 바로 할 수 있기 때문에 대량의 쓰기 부하 혹은 NVMM 접근 부하에서도 성능 유지가 가능하다.

이러한 특징들을 갖는 NOVA는 리눅스 커널 4.0에서 구현되었다. 또한 해당 커널에 있는 모든 NVMM 후크 (hooks)를 그대로 사용한다. 현재 리눅스 POSIX 파일시스템 테스트 수트 [17]를 통과하여 소스코드가 GitHub: https://github.com/NVSL/NOVA에 공개되어 있다. NOVA 파일시스템의 전체적인 구조, 원자성, 쓰기 순서 결정 메카니즘이 그림 1에 나타나 있다. NOVA는 확장성 제공을 위해 CPU마다 프리리스트, 저널, inode테이블을 갖는다. 각 inode는 4KB 로그 페이지의 단일 링크리스트로 구성된 분리된 로그를 갖는다. inode에서 끝 (tail) 포인터는 가장 최



(그림 1) NOVA 자료구조[1]

근에 반영된 로그 엔트리를 기록한다. 보다 세부적인 구현 상세는 [1]의 4장을 참조하기 바란다.

4. 결 론

하이브리드 메인 메모리를 위한 파일시스템은 기존의 메모리를 위한 것과는 다른 각 메모리 기술의 특징 및 장점을 충분히 살릴 수 있는 형태로 구성되어야 한다. 기존의 파일시스템은 새로운 메모리 기술에 대한 고려가 구현되어 있지 않기 때문에 그대로 적용하여 사용하면 성능이 저하되는 문제가 발생한다. NOVA는 가장 최근에 발표된 LFS로 하이브리드 메모리의 장점을 충분히 고려하여 설계되었다. 이러한 고려사항과 특징들은 향후 개발되는 하이브리드 메모리를 위한 파일시스템에서도 그대로 적용될 것으로 예상된다. 보다 단순하면서도 고성능을 제공할 수 있는 효과적인 가비지 콜렉션과 복구를 제공하는 기능은 모든 파일시스템이 제공해야 할 기본이기 때문이다. NVMM기반 하이브리드 메모리는 이러한 장점들로 인하여 조만간 메인메모리 시장을 크게 점유할 수 있을 것으로 예상된다. 다만 임베디드 시스템에 적용 가능한 하이브리드 메인 메모리에서는 이러한 기본 기능에 대한 요구조건이 크게 다를 수 있을 것이다. 다른 제약에 따라 복구 기능을 포기할 수도 있는 환경이기 때문이다. 이러한 환경에 적합한 파일시스템을 새롭게 고려하는 것도 향후 필요하다고 판단된다.

참 고 문 헌

[1] Jian Xu, Steven Swanson, “NOVA: A Log-structured File System for Hybrid Volatile/Non-volatile Main Memories”, 14th USENIX Conference on File and Storage Technologies (FAST ’16), 2016.

[2] P. Sehgal, S. Basu, K. Srinivasan, and K. Voruganti, “An Empirical Study of File Systems on NVM”, In Proceedings of the 2015 IEEE Symposium on Mass Storage Systems and Technologies (MSST’15), 2015.

[3] M. S. Bhaskaran, J. Xu, and S. Swanson, “Bankshot: Caching Slow Storage in Fast Non-volatile Memory”, In Proceedings of the 1st Workshop on Interactions of NVM/FLASH with Operating Systems and Workloads, INFLOW ’13, pp. 1:1–1:9, New York, USA, 2013.

[4] A. M. Caulfield, T. I. Mollov, L. A. Eisner, A. De, J. Coburn, and S. Swanson, “Providing safe, user space access to fast, solid state disks”, In Proceedings of the seventeenth international conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS XVII, pp. 387–400, New York, USA, 2012.

[5] J. Condit, E. B. Nightingale, C. Frost, E. Ipek, B. Lee, D. Burger, and D. Coetzee, “Better I/O through byteaddressable, persistent memory”, In Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles, SOSP ’09, pp. 133–146, New York, USA, 2009.

[6] J. Bonwick and B. Moore. ZFS: The Last Word in File Systems, 2007.

[7] C. Lee, D. Sim, J. Hwang, and S. Cho, “F2FS: A New File System for Flash Storage”, In 13th USENIX Conference on File and Storage Technologies, FAST ’15, pp. 273–286, Santa Clara, CA, Feb. 2015.

[8] J. Ousterhout, A. Gopalan, A. Gupta, A. Kejriwal, C. Lee, B. Montazeri, D. Ongaro, S. J. Park, H. Qin, M. Rosenblum, S.

Rumble, R. Stutsman, and S. Yang, “The RAMCloud Storage System”, ACM Trans. Comput. Syst., 33(3), pp. 7:1–7:55, Aug. 2015.

[9] J. Condit, E. B. Nightingale, C. Frost, E. Ipek, B. Lee, D. Burger, and D. Coetzee, “Better I/O through byte addressable, persistent memory”, In Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles, SOSP ’09, pp. 133–146, New York, USA, 2009.

[10] S. R. Dulloor, S. Kumar, A. Keshavamurthy, P. Lantz, D. Reddy, R. Sankaran, and J. Jackson, “System Software for Persistent Memory”, In Proceedings of the Ninth European Conference on Computer Systems, EuroSys ’14, pp. 15:1–15:15, New York, USA, 2014.

[11] M. Wilcox. Add support for NV-DIMMs to ext4. https://lwn.net/Articles/613384/.

[12] X. Wu and A. L. N. Reddy, “SCMFS: A File System for Storage Class Memory”, In Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, SC ’11, pp. 39:1–39:11, NY, USA, 2011.

[13] H. Volos, S. Nalli, S. Panneerselvam, V. Varadarajan, P. Saxena, and M. M. Swift, “Aerie: Flexible File-system Interfaces to Storage-class Memory”, In Proceedings of the Ninth European Conference on Computer Systems, EuroSys ’14, pp. 14:1–14:14, NY, USA, 2014.

[14] S. Chen and Q. Jin, “Persistent B+-trees in Non-volatile Main Memory”, Proc. VLDB Endow., 8(7), pp. 786–797, Feb. 2015.

[15] I. Moraru, D. G. Andersen, M. Kaminsky, N. Tolia, P. Ranganathan, and N. Binkert, “Consistent, Durable, and Safe Memory Management for Byte-addressable Non Volatile Main Memory”, In Proceedings of the First ACM SIGOPS Conference on Timely Results in Operating Systems, TRIOS ’13, pp. 1:1–1:17, NY, USA, 2013.

[16] Trees I: Radix trees. https://lwn.net/Articles/175432/.

[17] Linux POSIX file system test suite. https://lwn.net/Articles/276617/.

저 자 약 력

조 중 석

이메일 : icaroosion@naver.com

- 2006년~2012년 국립순천대학교 전자공학과 학사
- 2012년~2014년 국립순천대학교 전자공학과 석사
- 2014년~현재 국립순천대학교 전자공학과 박사과정
- 관심분야 : 임베디드 시스템, 저전력 메모리 시스템, 시스템 소프트웨어, 머신러닝

조 두 산

이메일 : mew26@snu.ac.kr

- 2010년~2012년 국립순천대학교 전자공학과 전임강사
- 2012년~2014년 국립순천대학교 전자공학과 조교수
- 2014년~현재 국립순천대학교 전기전자공학부 부교수
- 관심분야 : 최적화 컴파일러, 임베디드 시스템, 저전력 메모리 시스템, 시스템 소프트웨어, 머신러닝, 인공지능

특집 03

Key-Value Store를 위한 저장 매체 기술 연구 동향

임승호 (한국외국어대학교)

목 차

1. 서 론

Key-Value Store 또는 Key-Value Database는 key-value 쌍으로 이루어진 데이터 구조 배열 값을 저장, 검색, 관리하는 데이터 저장 기술 패러다임이다[1,2]. 일반적으로 key-value로 관리되는 데이터 구조는 key값 해당하는 dictionary 또는 hash값에 대해서 value를 저장하는 배열 형식의 구조를 가진다. Key-Value Store는 기존의 데이터베이스 시스템에 널리 사용되는 관계형 데이터베이스(RDB)와는 매우 다른 방식으로 동작한다. 관계형 데이터베이스는 잘 정의 된 데이터 유형 필드를 포함하는 테이블의 일련의 데이터베이스의 데이터 구조를 미리 정의하기 때문에 때로는 무겁고 성능에 제약이 따를 수 있다. 이와는 대조적으로, Key-Value Store는 각 레코드에 대한 다양한 필드를 가질 수 있는 하나의 불투명 컬렉션과 같은 데이터를 취급하는데, 이것은 상당한 유연성과 단순성을 제공한다.

이러한 유연성으로 인해서 Key-Value Store는 인터넷의 넘쳐나는 데이터를 저장하고 검색하고 관리하는 빅데이터 시스템의 저장 구조로 널리 사용되고 있다. Key-Value Store기반의 빅데이터 시스템은 high-throughput과 low-latency를 제공함으로써 이를 요구하는 다양한 많은 인터넷 응용 서비스에 활용되고 있다. Key-Value Store는 인터넷 기반의 빅데이터 시스템의 특성상 많은 데이터의 저장, 검색, 관리에 사용되기 때문에 key-value를 저장하는 저장장치 기술이 주요한 기술 중 하나라 볼 수 있으며, 고속의 검색과 고가용성을 보장하기 위해서 저장 매체를 어떤 방식으로 사용하느냐가 최근의 저장매체의 발달과 함께 주요 이슈라고 볼 수 있다.

Key-Value Store의 저장매체 관련 연구는 전통적인 저장 매체인 HDD 기반의 저장 매체 활용 기술에서부터, In-memory 기반의 key-value 관리 방법, 그리고 최근에 각광받고 있는 Flash Memory기반의 저장매체를 기반으로 성능 향상

연구를 비롯하여 다양한 연구 및 기술개발이 페이스북, 트위터, 구글과 같은 글로벌 인터넷 기업에서 진행 중에 있다[6,7,8,9,10,17,18]. 본 고에서는 Key-Value Store의 저장매체 기술에 대한 최근 연구 동향과 글로벌 기업들이 자사의 제품에 직접 적용한 기술 사례들을 살펴보고 향후 Key-Value Store 관련 저장 매체 적용 기술이 나아갈 방향에 대해서 모색해 보도록 한다.

2. Key-Value Store 연구동향

2.1 NoSQL의 Key-Value Store

Key-Value 데이터 모델은 빅데이터 시스템에서 널리 사용되는 NoSQL의 기본 데이터 모델이다. Key-Value Store 모델은 그 구성방식에 따라서 단순한 Key-Value 형태의 데이터 모델, Document Oriented 데이터 모델, Column Family Oriented 데이터 모델 등 다양한 형태의 데이터 모델로 확장될 수 있는데, 이들이 RDBMS형식의 데이터베이스와는 다른 주요한 특징은 단순성, 가용성, 확장성, 그리고 속도적인 측면이 있다. RDBMS의 주요 특징인 ACID(원자성, 일관성, 고립성, 지속적인 트랜잭션)을 지원하는 대신에 BASE(기본적 가용성, 소프트 상태, 결과적 일관성)을 제공하는 것을 주요 목적으로 한다. 이러한 측면으로 인해서 Key-Value Store는 서버 여러 대로 이루어진 클러스터 환경에서 주로 사용된다.

Key-Value Store는 여러 가지 형식을 가진 데이터 모델이 있지만, 기본적으로 Key-Value 쌍으로 이루어진 수많은 Row 값을 배열 형태로 기록하는 데이터 모델이다. 많은 Key-Value Store에서는 이러한 데이터 모델을 기반으로 인터넷 응용 시스템에서 넘쳐나는 데이터를 신속하게 저장하고, 저장된 대용량의 Key-Value의 검색의 가용성 및 신속성을 보장하기 위해서, 로그 형식의 데이터 기록 및 저장 방식과, Hash기반 테이블의 분산 저장, In-Memory Cache 구조를 가진다.

2.2 Key-Value Store 종류

Key-Value Store는 초창기 임베디드용으로 시작하여 SQL을 단순화한 Key-Value Store부터 최근의 빅데이터 및 대용량 인터넷 서비스를 위한 디스크 기반 분산 데이터베이스와 메모리 기반 분산 데이터베이스 등 다양한 종류의 Key-Value Store가 있다. 현재 빅데이터 시스템에서 가장 널리 사용되는 Key-Value Store의 대표적인 예로는 BerkeleyDB[3], HashCache[4], Memcached[5], BigTable[6], DynamoDB[7], Cassandra[8], LevelDB[9], HBase[10]등이 있으며, 여기서는 주요한 Key-Value Store를 소개한다.

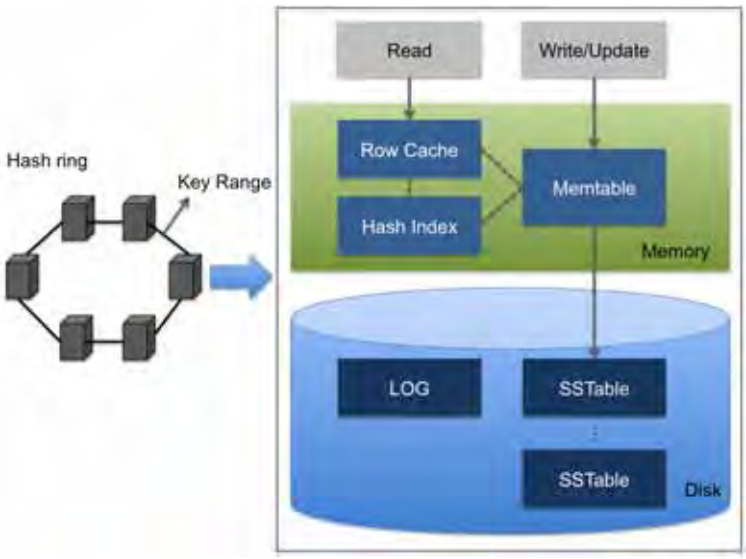
BerkeleyDB[3]는 1992년경 UCB에서 개발한 임베디드 Database이다. 유닉스 프로그램에서 보통 DB라 하면 이 버클리 DB를 말할정도로 대중적인 임베디드 데이터베이스로 알려졌으나, BerkeleyDB는 관계형 데이터베이스가 아니다. 임베디드용으로 활용되기 위해서 SQL의 복잡도와 무게를 제거하였으며, BerkeleyDB 저장 방식은 임의의 Key-Value의 쌍을 바이트 배열로 저장하는 방식이므로 초창기 Key-Value Store라고 할 수 있다. 그러나 Oracle Berkely DB 11g Release 2 버전부터 SQL과 SQLite API 를 지원하게 되었으며, 락이나 트랜잭션을 지원하며 Join나 Replication, In-Memory 데이터베이스도 지원하도록 진화되었다. BerkeleyDB에 SQL 분석기를 붙이고 인덱스 구성 가능하게 만든 것이 초기의 MySQL이다.

BitTable[6]은 구글 파일 시스템(Google File

System) 기반으로 개발된 고성능 Key-Value Store이다. BitTable은 분산 데이터베이스이며 Bigtable은 Row Key와 Column Key의 두 임의의 문자열 값과 timestamp값의 배열을 메모리에 테이블 형태의 tablet으로 기록한다. 각각의 테이블은 timestamp에 따라서 다수의 버전을 가지며 Google File System에 최적화 될 수 있도록 다수의 tablets으로 구성되며 200MB 크기를 가질 수 있다. tablet이 가득 차게 되면 GFS에 파일 형태의 로그를 기록한다. HBase[10]는 구글의 Bigtable의 기본 데이터모델을 기반으로 하여 Hadoop 그룹에서 개발한 HDFS(Hadoop Distributed File System)기반의 분산 저장 시스템을 활용한 Key-Value Store이다.

Cassandra[8]는 구글의 BigTable[6]과 아마존의 DynamoDB[7]를 기반으로 페이스북에서 개발한 분산 Key-Value Store이며 향후 Apache 오픈소스 프로젝트로 진행되었다. Apache Cassandra는 지속적인 가용성, 높은 확장성 및 성능, 강력한 보안 및 운영 단순성을 제공함으로써 현재 인터넷 빅데이터 시스템에서 가장 널리 사용되는 Key-Value Store 관리 시스템이라고 할 수 있다. 그림 1은 Cassandra Key-Value Store의 전형적인 구조를 도식화한 것이다. Cassandra의 데이터 모델은 구글의 BigTable기반으로 하였으며, 메모리상에 Memtable 구조의 Key-Value 어레이 값을 유지하며, Memtable의 Log형식의 값을 디스크상에 SSTable 구조로 저장한다. Cassandra의 분산 시스템의 구조는 아마존의 DynamoDB를 따르는데, Distributed Hash Table형식의 Hash Ring구조의 분산 시스템을 구성한다.

Memcached[5]는 데이터베이스 기반의 웹 서비스 시스템에서 스토리지에 저장되어 있는 데이터베이스 소스데이터를 읽는 부하를 줄이기 위해서 데이터베이스 오브젝트를 RAM에 Caching하여 관리할 수 있도록 만들어진 분산 메모리 캐싱 시스템으로 개발되었다. Memcached는 In-memory 분산DB를 관리하기 위해서 client-server 구조를 사용한다. 서버는 Key-Value



(그림 1) Cassandra 데이터 모델

관계에 대한 연관 배열을 유지하고 관리하며, Client는 이 배열의 값을 채우고 key에 대한 query를 담당한다. Memcached 자체는 메모리 상에서만 Key-Value값을 유지하기 때문에 Key-Value값을 저장하는 서버에서 메모리가 다 차게 되면 이전의 값을 버리는 방식으로 메모리에 최신 Key-Value값을 유지한다. Memcached의 단점을 보완 및 확장하여 디스크 기반의 저장까지 가능하게 만든 것이 MemcacheDB인데, MemcachedDB의 디스크 부분 관리는 BerkeleyDB [3] 구조를 사용한다.

3. Key-Value Store와 저장매체

3.1 Disk-Optimized Key-Value Store와 In-Memory Caching

Key-Value Store기반의 빅데이터 시스템에서도 실시간으로 급격히 생성되는 데이터의 저장 관리 데이터 규모가 테라(Tera)바이트를 넘어서 페타(Peta)바이트 단위로 돌입한 지 오래이다. HDD 기반의 디스크 저장장치는 값싼 가격 대비 대용량의 장점이 있지만, 기계적인 플래터와 헤더를 사용하는 구조적 한계로 인해서 임의(Random) 읽기/쓰기가 순차(Sequential) 읽기/쓰기에 비해서 급격히 느리다는 단점이 있다. Key-Value Store에서는 이러한 단점을 보완하기 위해서 읽기와 쓰기를 구분하여 다른 형식의 연산을 가지도록 구현하고 있다. 이는 GFS, HDFS, HFS등 기존의 대용량 분산 파일 시스템과 같은 방식인데, 기본적으로 기록을 위해서 생성된 요청연산(Requests)는 로그 형식으로 데이터를 순차적으로 기록하는 방식을 사용하며, 읽기를 위해서 생성된 요청연산(Requests)는 메모리 캐쉬를 이용하는 방식이다. 그렇기 때문에 기본적인 저장장치의 용량이 테라(Tera)~페타(Peta) 바이트의 용량을 가지며, Key-Store Caching을 위한 메모리는 수십기가~수백기가바이트의 용량을 가진다. 초창기 Bigtable, Cassandra, MemcacheDB, MongoDB등이 이러한 구조로 동작하는 Key-Value Store라 할 수 있다.

Cassandra의 예를 들어보면, Cassandra의 모든 쓰기(Write/Update) 요청 연산은 대용량 메모리의 Key-Value 데이터배열인 Memtable에 순차적으로 기록된다. Memtable에 지정된 용량에 도달했을 경우, Memtable은 디스크에 Log 데이터 형식으로 저장되며 이를 SSTable에 기록되는 방식이다. Memtable의 크기가 수 MB에서 수십 MB의 크기로 디스크에 Log 형식으로 기록되기 때문에 디스크에 순차적인 쓰기 연산을 수행할 수 있는 것이다. 반면 읽기 요청 연산의 경우 메모리상의 Row Cache와 Memtable을 검색하여 해당 데이터를 존재하는 경우 읽기 요청 연산을 반환한다. 만약 메모리에 해당 요청 데이터가 없을 경우 디스크 저장장치의 SSTable을 검색하여 원하는 Key에 대한 Row를 구성하여 반환하는데, 이 때 다수의 SSTable에 존재하는 버전을 검색하여 Row를 구성해야 하기 때문에 다수의 랜덤 읽기가 발생할 수 있다.

Key-Value Store 저장 시스템에서 읽기/쓰기 연산의 비대칭으로 인해서 발생하는 또다른의 문제점은 업데이트되는 데이터의 Log 기록 연산에 의해서 발생하는 가비지(garbage)처리를 위한 compaction 또는 cleaning 작업이다. 특히 Key-Value Store는 row당 timestamp버전을 관리하기 때문에 이러한 compaction작업으로 인해서 내부적으로 처리해주는 쓰기 연산 오버헤드가 실제적인 외부 요청에 의한 쓰기 연산에 비해서 수배~수십배의 쓰기 오버헤드가 발생하는 문제점이 있다.

3.2 Flash Disk의 Evolution

플래시 메모리, 특히 NAND 플래시 메모리의 급격한 대용량화는 저장장치 최근 수년간 저장장치 시장의 패러다임 뿐만 아니라 컴퓨터 시스템 구조의 급격한 변화를 가져오고 있으며, 낸드 플래시 메모리 기반의 저장장치인 SSD(Solid State Disk)의 최근의 급격한 성능 향상과 대용량화는 전통적인 HDD를 대체하기에 이르고 있다. 낸드 플래시 메모리의 물리적인 특징으로 인해서 SSD의 순차 읽기/쓰기 연산속도 뿐만 아니라 임의 읽기/쓰기 속도가 HDD에 비해서 급격히 개선되고 있다.

낸드 플래시 메모리는 플로팅 게이트를 사용하는 반도체소자로써. 낸드 플래시 메모리의 기본적인 읽기쓰기 연산의 크기는 페이지(page)라는 단위로써 크기가 4KB~8KB크기이다. 쓰기 페이지에 쓰기 연산을 수행하기 위해서는 지우기(Erase)연산이 먼저 수반되어야 하는데, 그 크기는 블록(Block)이라는 단위이며 64~256개의 페이지그룹이다. 또한 블록 지우기 연산의 횟수는 블록당 제한이 있는데 집적도를 높이여서 용량을 점점 증대시킬수록 이 횟수는 점점 줄어든다. 인해서 낸드 플래시 기반의 SSD등 저장장치 내부에 플래시 변환 계층(Flash Translation Layer)라는 소프트웨어 또는 펌웨어가 물리적인 제약을 보완해 준다. 플래시 변환 계층의 주요한 기능으로는 호스트 논리 주소와 낸드 플래시 메모리의 물리주소 변환 관리, 웨어 레벨링 관리, 가비지 콜렉션등의 작업을 수행한다.

SSD와 같은 Flash Disk는 최근의 급격한 대용량화와 가격 하락으로 인해서 대용량 서버 시스템에서 그 점유율을 점점 더 높여가고 있으며, 서버의 성능하락의 주요 요인이었던 저장장치 IO의 한계를 극복할 수 있는 여건을 마련해주고 있다. 특히 임의 읽기/쓰기 연산 속도의 급격한 향상으로 인해서 기존 HDD가 가지고 있는 문제를 해결할 수 있는 대안으로 각광받고 있다. Key-Value Store 시스템에 플래시 디스크 저장장치를 활용하는 방법은 기존의 Memory-Disk 저장장치 기반에서 Flash Disk를 어떤 식으로 조합하느냐의 문제일 수 있다. Flash Disk 저장장치를 Key-Value Store에 적용한 기술 개발에는 IBM의 CaSSanDra[16], 페이스북의 McDipper [17], 트위터의 Fatcache[18], 그리고 NVMKV [20]등이 있다.

SSD boosted CaSSanDra[16]는 기존의 Cassandra Key-Value Store 시스템의 Memory -HDD사이에 SSD를 추가하여 Hybrid HDD/SSD를 활용한 Memory-SSD-HDD 저장장치 구조를 제안하고, SSD를 Second Level Cache로 활용한 연구를 진행하였다. SSD boosted CaSSanDra는 Key-Value Store에 저장되는 각 row의 column명들이 거의 변하지 않음에도 불구하고 SSTable에 저장되는 것이 중복되는 내용이며, HDD의 Ramdom I/O의 성능이 안좋은 영향을 미치는 것에 착안하여, Key-Value Store에 저장되는 Column 명들을 따로 Schema Catalogue라고 하는 것으로 분리하여 SSD에 Caching하는 Key-Value Store Architecture를 제안하였다.

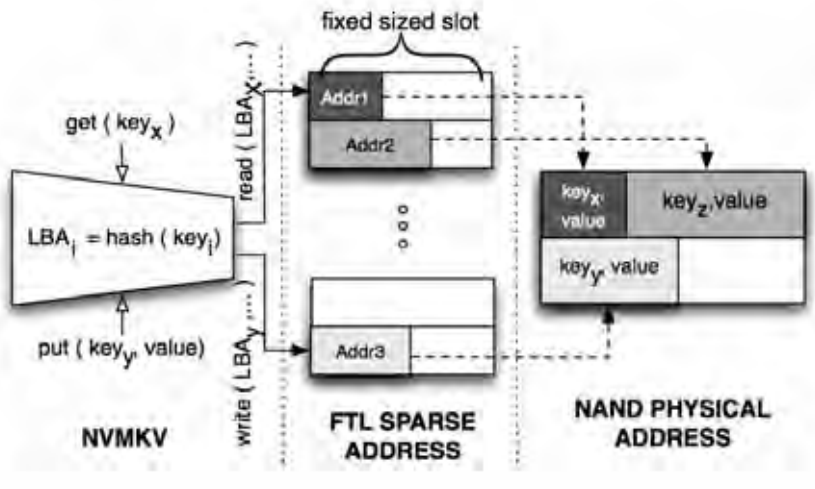
MCDipper[17]는 페이스북에서 In-memory Key-Value Store 캐시인 Memcache를 SSD 디바이스 기반으로 재설계 및 구현한 Key-Value Store이다 Memcache를 위해서 수십기가~수백기가바이트의 RAM이 필요한데, 이는 성능 향상에는 좋지만 고비용의 시스템 구성이 필요하다. 페이스북에서는 RAM보다는 값싸지만 초당 수십만 IOPS(IO Operations Per Second)의 성능을 보장하는 SSD로 RAM을 대체할 경우 가격효율

적인 Key-Value Store 캐쉬를 구현할 수 있다고 보았다. Memcache를 SSD기반으로 재구성한 프로젝트는 트위터에서 진행된 FatCache[18]도 있다. 메모리 시스템에 최적화된 Memcache 구현을 SSD기반으로 안정적으로 동작하도록 구현하는 것이 쉬운 일은 아니지만, MCDipper와 Fatcache는 SSD의 물리적인 특징을 설계시 고려하지 않았다는 점에서 한계점이 있을 수 있다.

FusionIO와 관련 연구그룹에서는 NVMKV [20] 연구결과를 발표하였는데, 이는 SSD의 낸드 플래시 메모리의 특성 및 FTL의 기능을 Key-Value Store의 API로 활용하여 Key-Value Store의 저장장치 성능을 향상시키기 위한 시도였다. 그림 2는 NVMKV의 Key-Value Hash Model을 도식화한 것으로써, NVMKV는 FTL의 주요 특징인 다이나믹 매핑 관리, 로그 구조의 데이터 기록, 트랜잭셔널 데이터 업데이트, 씬 프로비져닝(thin provisioning) 등이 Key-Value Store에서 사용되는 저장장치 관리 기법과의 유사점이 있다고 보고, FTL의 API를 유저레벨 Key-Value Store의 API로 확장하여 Key-Value Store관리에 사용할 수 있도록 라이브러리 형태로 제공하도록 구현하였다. 이를 통해서 Flash 최적화된 Key-Value Store 구성이 가능하다.

최근의 Open-Channel SSD[21]를 기반으로 한 SSD-transparent Key-Value Store 설계 연구는, NVMKV에서 더 나아가서 플래시 디스크의 낸드 플래시 메모리 관리 부분까지 Key-Value Store에서 운영 및 관리하고자 하는 시도이다. Open-Channel SSD는 기존의 SSD와는 다르게 FTL과 같은 플래시 소프트웨어를 운영체제 내에서 직접 관리하는 SSD 구조를 말한다. Key-Value Store에서 Open-Channel SSD 기반의 저장장치 운영 및 관리를 하게 될 경우의 가장 큰 이점은 기존의 SSD 기반 Open-Channel에서 발생하는 가장 큰 문제점인 Write Amplification을 줄여줄 수 있으므로 쓰기 최적화된 Key-Value Store 구성이 가능하다는 장점이 있다.

이상과 같이 플래시 디스크 기반의 Key-Value Store는 지난 수년간 SSD의 급속한 발전과 더불어 함께 진행되어 오고 있으며, 기존의 HDD기반의 저장장치 구조에서 탈피하여, Memory, SSD,



(그림 2) NVMKV의 Key-Value Hash Model[20]

HDD를 다양한 측면에서 적용하여 기술 진화를 이룩하였다. 향후에도 이러한 저장장치 관리 및 운영의 연구를 통해서 Key-Value Store 및 빅데이터 시스템의 진화는 계속 될 것으로 보인다.

4. 결 론

Key-Value Store는 NoSQL 데이터베이스의 데이터 저장 방식으로써 기존의 관계형 데이터베이스에 비해서 비정형화되어 단순성, 가용성, 확장성, 그리고 처리 속도에서 성능이 상대적으로 우수하여 빅데이터 시스템의 데이터 저장 및 검색에 널리 사용되고 있다. Key-Value Store의 대표적인 예로는 구글의 Bigtable, LevelDB등이 구글 클러스터 서버와 구글 파일 시스템의 빅데이터 저장 구조로 사용되며, 페이스북의 Cassandra, 아마존의 Dynamo, Hadoop의 HBase등이 있다.

이러한 Key-Value Store는 지속적으로 늘어나는 데이터의 저장 및 검색의 효율성을 위해서 저장장치 구조 및 동작은 Key-Value의 어레이를 In-Memory Cache에 기록한 후, 일정 크기의 로그를 디스크에 기록하는 쓰기/업데이트 연산과, In-Memory Cache 검색후, Cache에 존재하지 않는 경우 다수의 임의 읽기를 통해서 Key-Value Row를 구성해서 반환하는 읽기 연산으로 구성하며, Key-Value에 대한 분산 Hash를 적용하여 분산 데이터베이스 시스템을 구성한다.

Memory-HDD기반의 Key-Value Store는 구조상 빅데이터 저장장치 시스템의 성능 및 확장이 비효율적인 측면이 많이 나타났으며, 이를 극복하기 위해서 최근의 플래시 메모리 기반의 SSD가 Key-Value Store의 저장장치 시스템에 많이 도입되어 가격 및 성능 향상의 역할을 하고 있다. 향후 SSD는 집적도 향상으로 비용이 더욱 더 저렴해질 것이며, NVMe PCIe[22] 인터페이스와 같은 호스트 인터페이스의 진화, Open Channel SSD등 플래시 소프트웨어의 다변화로 인해서 더욱더 Key-Value Store를 비롯한 빅데이터 및 인터넷 서버 시스템에서의 활용이 증대될 것으로 기대된다.

참 고 문 헌

[1] J.Manyika, M.Chui, B.Brown, J.Bughin, R.Dobbs, C.Roxburgh, and A. H. Byers, “Big data: The Next Frontier for Innovation, Competition, and Productivity,” McKinsey Global Institute, Tech. Rep., 2011.

[2] T. Rabl, M. Sadoghi, H.-A. Jacobsen, S. Gómez-Villamor, V. Munte´s- Mulero, and S. Mankowskii, “Solving Big Data Challenges for Enter- prise Application Performance Management,” PVLDB, 2012.

[3] M. Olson, K. Bostic, and M. Seltzer. Berkeley db. In Proc. of USENIX Annual Technical Conference, 1999

[4] A. Badam, K. Park, V. Pai, and L. Peterson. Hashcache: cache storage for the next billion. In Proc. of USENIX Networked System Design and Implementation, 2009.

[5] memcached - a distributed memory object caching system https://memcached.org/

[6] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, “Bigtable: A distributed storage system for structured data,” in OSDI, 2006.

[7] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, “Dynamo: Amazon’s Highly Available Key-Value Store,” in SOSP, 2007.

[8] A. Lakshman and P. Malik, “Cassandra: a decentralized structured storage system,” SIGOPS Review, 2010.
[9] LevelDB – a fast and lightweight key/value database library by Google. http://code.google.com/p/leveldb/.
[10] Apache HBase. http://hbase.apache.org/.
[11] M. Sadoghi, K. A. Ross, M. Canim, and B. Bhattacharjee, “Making updates disk-I/O friendly using SSDs,” PVLDB’13.
[12] R. Cartell, “Scalable SQL and NoSQL data stores,” SIGMOD Record, 2010.
[13] M. Cornwell, “Anatomy of a solid-state drive,” Communications of the ACM, 2012.
[14] L. Bouganim, B. r Jnsson, and P. Bonnet, “uFLIP: Understanding Flash IO Patterns,” in CIDR ’09: Fourth Biennial Conference on Innovative Data Systems Research.
[15] G. Graefe, “The Five-Minute Rule 20 Years Later: and How Flash Memory Changes the Rules ,” Communications of the ACM, 2009.
[16] Prashanth Menon, Tilmann Rabl, Mohammad Sadoghi, Hans-Arno Jacobsen, “CaSSanDra: An SSD Boosted Key-Value Store”, IEEE ICDE Conference, 2014.
[17] MCDipper, https://www.facebook.com/notes/facebook-engineering/mcdipper-a-key-value-cache-for-flash-storage/10151347090423920
[18] Fatcache, https://github.com/twitter/fatcache
[19] LIM, H., FAN, B., ANDERSON, D. G., AND KAMINSKY, M. SILT: A Memory-Efficient, High-Performance Key-Value Store. In Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP’11) (Cascais, Portugal, October 23-26 2011)
[20] MA ´RMOL, L., SUNDARARAMAN, S., TALAGALA, N., AND RAN- GASWAMI, R. NVMKV: A Scalable and Lightweight, FTL-aware Key-Value Store. In Prceedings of USENIX ATC’15 (2015).
[21] Open-Channel Solid State Drives, http://openchannelssd.readthedocs.io/en/latest/
[22] NVMe (Non-Volatile Memory Express), http://www.nvmexpress.org
[23] YANG, J., PLASSON, N., GILLIS, G., TALAGALA, N., AND SUN- DARARAMAN, S. Don’t Stack Your Log on My Log. In Proceedings of the 2nd Workshop on Interactions of NVM/Flash with Operating Systems and Workloads (INFLOW’14) (Broomfield, CO, October 5 2014).
[24] ZHANG, Y., ARPACI-DUSSEAU, A. C., AND ARPACI-DUSSEAU, R. H. De-indirection for Flash-based SSDs with Nameless Writes. In Proceedings of FAST’12 (2012).
[25] ZHANG, Y., ARPACI-DUSSEAU, A. C., AND ARPACI-DUSSEAU, R. H. Removing the Costs and Retaining the Benefits of Flash-Based SSD Virtualization with FSDV. In Proceedings of MSST’15 (Santa Clara, CA, June 2015).
[26] L. Grupp, J. Davis, and S. Swanson. The bleak future of nand flash memory. In Proc. of USENIX FAST, 2012.

임 승 호

이메일: slim@hufs.ac.kr

- 2001년 한국과학기술원 전기 및 전자공학과 (학사)
- 2003년 한국과학기술원 전기 및 전자공학과 (석사)
- 2008년 한국과학기술원 전기 및 전자공학과 (박사)
- 2008년~2010년 (주)삼성전자 메모리사업부 책임연구원
- 2010년~현재 한국외국어대학교 컴퓨터.전자시스템공학부 부교수
- 관심분야 : 운영체제, 임베디드 시스템, 파일 시스템, 플래시 메모리, 빅데이터 저장장치

특집 04

클라우드 스토리지 보안 기술 동향

황우민 · 김성진 · 김형천 (국가보안기술연구소)

목 차

1. 서 론

클라우드 컴퓨팅은 주문형 셀프 서비스, 광대역 네트워크 접근, 자원 공동관리, 빠른 탄력성, 측정 가능한 서비스의 특성을 갖는 차세대 컴퓨팅 구조[58]로서 가용성, 유연한 확장성, 높은 접근성 등의 장점을 앞세워 빠르게 확산되고 있다. 사용자들은 클라우드에 저장된 자신의 데이터를 위치에 구애받지 않고 클라우드로의 네트워크 연결만으로 접근할 수 있으며 스토리지 관리 부담을 줄일 수 있고 필요한 만큼 사용하고 그에 대한 비용만을 지불할 수 있기 때문에 비용 절감 효과를 얻을 수 있는 장점이 있다. 하지만 클라우드 환경에서는 다수의 사용자가 중앙 집중화된 컴퓨팅 환경을 공유하기 때문에 저장된 데이터를 보호하기 위한 기본적인 데이터 안전성 확보와 자료 유출 방지, 복구, 모니터링 및 통합 인증, 접근 권한 관리 기술 등 다양한 보안 요구사항을 만족시킬 수 있는 확장성 있는 저장기술이 필요하다.

주요 국제 기관들에서는 이러한 클라우드 컴퓨팅 환경에서 발생할 수 있는 보안 위협에 대해서 분류, 분석하고 보안 요구사항들을 도출하는 작업을 수행하고 있다. CSA(Cloud Security Alliance)는 클라우드 환경에서의 보안 위협을 12가지로 분류하여 분석하였고[1], ITU-T에서는 X.1601에서 클라우드 컴퓨팅 보안 위협을[7], ENISA는 17종류의 사이버 위협들을 분류하고 각각에 대해서 분석하였다[8].

본 고에서는 이들 중 클라우드 스토리지의 보안과 관련된 네 가지 위협, 즉, 데이터 유출, 유실, 악의적인 서비스 제공자 위협 및 공유 기술 문제를 간략히 정리하고 클라우드 스토리지와 관련된 보안 기술 연구 동향에 대하여 기술한다.

2. 클라우드 스토리지

클라우드 스토리지는 클라우드 환경에서 데이터가 저장되는 논리적 스토리지 연합체로써 다

수의 분산된 저장 장치를 연결하여 저장 공간을 생성함으로써 클라우드 사용자로 하여금 데이터를 저장하고 접근할 수 있도록 관리하는 스토리지로 정의된다. 이러한 클라우드 스토리지에 사용되는 기술의 주요 예로써 HDFS, Swift, Glory-FS, OwFS, GlusterFS, NFS/CIFS, 상용 서비스의 예로써 Google Cloud Storage, Amazon S3 등이 있다.

HDFS(Hadoop Distributed File System)[10,11]는 대량의 자료 처리를 위한 분산 응용 프로그램을 지원하는 Hadoop 분산처리 프레임워크의 분산 파일 시스템으로 커버로스(Kerberos) 및 토큰 시스템을 이용한 인증 및 접근 제어를 사용하며 이와 유사한 Glory-FS[9]도 데이터 접근 제어 및 암/복호화를 지원한다.

Swift[12]는 OpenStack의 object storage 시스템으로서 분산된 상용 스토리지 자원들을 이용하는 스토리지 서비스로 오류를 대비하여 객체들을 클러스터 내 저장장치에 중복되어 분산 저장하고 데이터의 무결성을 제공한다. 안정화 버전에서는 Keystone을 이용한 권한관리만을 제공하지만 개발 버전에서는 AES 암호화를 적용 중이다. 네이버가 개발하여 사용하고 있는 대규모 분산 파일 시스템인 OwFS(Owner-based File System)[13]는 파일에 대한 암호화를 적용하고 있고 GlusterFS[14]는 파일의 내용에 대한 암호화 기능과 함께 데이터로의 접근시 사용자 인증 기능도 함께 제공하고 있다. NFS[15]/CIFS는 네트워크를 통해 다른 호스트에 있는 파일을 공유해서 사용할 수 있는 분산 파일 시스템으로 로컬 파일 시스템과 동일한 기능을 제공하므로 응용 프로그램의 구동시 로컬 파일 시스템과 분산 파일 시스템을 구별하지 않는다. NFS는 version 4에서 클라이언트/서버 인증과 Kerberos를 통해서 데이터 무결성/기밀성을 제공한다.

이외에도 상용 서비스들, 예를 들어 Google Cloud Storage[5]는 모든 데이터에 대해 쓰기 작업이 진행되기 전 256bit AES 암호화를 강제하는 정책을 적용중이고 Amazon S3[39]는 클라이언트/서버측에서 암호화를 지원하여 서비스 사용자 및 제공자 측면에서의 데이터 암호화를 제공한다.

이러한 예에서 볼 수 있듯 대다수의 클라우드 스토리지가 데이터 암호화 및 접근 사용자의 인증을 제공하지만 저장 데이터의 대형화를 수용하기 위해 각 구성 요소들이 네트워크화되고 있어 이에 적합한 암호화 기술 및 접근 제어 기술의 필요성이 증가하고 있다. 하지만 대용량 데이터 처리에의 암호화/접근 제어 기술의 적용은 성능 저하를 야기할 가능성이 높으므로 클라우드 스토리지에 저장되는 데이터를 효율적으로 보호하기 위해서는 연산 효율적 암호화 기술의 개발 및 적용이 필요할 뿐만 아니라 보다 정확한 위협의 식별 및 이에 따른 데이터로의 접근 통제 및 보호를 위한 기술 개발, 저부하 데이터 흐름 모니터링 기술 등의 연구가 필요하다.

3. 클라우드 스토리지 보안 기술 연구 동향

3.1 보안 위협 요소

클라우드 컴퓨팅과 관련하여 2016년도에 CSA가 갱신한 12가지 위협 요소 분류, ITU-T X.1601[7]에서 분류한 클라우드 컴퓨팅 보안 위협, ENISA에서 분류한 17가지 사이버 위협들[8] 중에서 클라우드 스토리지와 관련성이 높은 것들을 [6]에서 식별한 과거의 보안 위협을 참조하여 CSA의 용어를 기준으로 분류하면 1) 데이터 유출; 2) 악의적인 서비스 제공자; 3) 데이터 유실;

4) 공유 기술 문제의 네 가지로 나타낼 수 있다.

3.1.1 데이터 유출

2016년도에 갱신된 클라우드 스택[2]의 보안 부분에도 본 위협에 대한 대비가 요소 기술의 하나로 포함되어 있는 데이터 유출 위협은 허가받지 않은 사용자의 데이터 생성, 접근, 사용에 대한 통제가 적절하게 이루어지지 않을 경우에 존재한다. 공유를 기반으로 하는 클라우드 환경에서 저장된 데이터는 내부의 임의 사용자, 같은 스토리지를 사용하는 임의의 가상 머신, 네트워크를 통한 접근자 등에 의해서 접근될 수 있다. 특히 IaaS (Infrastructure as a Service) 및 PaaS (Platform as a Service) 서비스 모델을 사용하는 환경은 기반이 되는 클라우드 인프라에 대한 접근성이 높은 특성을 가지므로 유출 위협으로부터 데이터를 보호하기 위한 추가적인 노력이 필요하다. 예를 들어, 금융권 및 의료 관련 데이터와 같이 극도의 기밀성을 유지해야 하는 데이터의 경우 암호화 및 높은 접근 통제 수준을 적용해야 한다.

3.1.2 악의적인 서비스 제공자

기존 내부자 보안은 기업 내부에 통제권이 있기 때문에 높은 통제 수준을 유지할 수 있었지만 기업 내부 데이터들이 클라우드 환경에 존재하는 경우 이를 관리하는 통제권이 클라우드 서비스 제공자에게 있게 된다. 기업은 자신의 데이터를 관리하는 사람에 대한 통제권이 약해지게 될 수 밖에 없어 클라우드 서비스 제공자 내부의 악의적인 관리자가 기업의 데이터에 대한 접근 시이를 통제하고 모니터링할 수 있는 방안이 필요하며 이러한 보안 정책이 이행되고 있는지를 관리, 감독할 수 있어야 한다.

3.1.3 데이터 유실

데이터 유실 위협은 2009년도 MS 사이드킥 사고[3], 2011년도 구글 Gmail 삭제 사고, 2015년 구글 데이터 유실 사고[4] 에서와 같이 실존하면서도 지속적으로 발생하는 보안 위협이다. 2014년도 6월에는 온라인 호스팅 업체인 Code Spaces가 해킹에 의한 사용자 데이터의 파괴로 서비스를 종료한 사례[1]에서도 볼 수 있듯 데이터 손실에 대비하기 위한 다양한 방법이 존재하지만 이들이 정상적으로 동작하지 않거나 대비하지 못한 유형의 원인에서 발생하는 위협일 경우 큰 피해가 발생할 수 있다. 따라서 다양한 유형의 원인으로부터 야기되는 손실에 대비할 수 있는 안정적이면서도 효율적인 저장기술이 필요하다.

3.1.4 공유 기술 문제

클라우드의 확정성은 클라우드 인프라를 공유함으로써 달성된다. 저장 공간을 비롯한 공유되는 자원 상에서 이용자들에 대한 서비스가 이루어지기 때문에 가상 환경의 바탕이 되는 공유 자원 상에서 강력한 격리가 이루어지지 않을 경우 임의의 사용자가 다른 사용자의 데이터에 접근한 수 있는 상황이 발생할 수 있다. 클라우드 서비스 제공자는 적절한 수준의 공유 기술을 적용함으로써 효율성과 확장성을 보장하는 동시에 물리적/논리적 격리 기술을 적용하여 공유 기술에 의한 보안 위협을 방지해야 한다.

이외에도 [6]에서는 서비스 가용 여부를 저장장치 가상화 시스템 보안 위협의 하나로 보고 사고 모니터링 및 사고 대응을 보안 요구사항의 하나로 제안하고 있다.

3.2 스토리지 보안 기술 동향

이전 절에서 언급한 보안 위협에 대응하기 위해서 다양한 기술에 대한 연구가 진행되고 있다. 클라우드 컴퓨팅 환경에서의 스토리지 보안 기술은 크게 사용자 측면의 클라우드 단말 보안 기술과 서비스 제공자 측면의 클라우드 서비스 보안 기술로 나눌 수 있다. 사용자 단말 측면에서의 스토리지 보안 기술에는 단말 스토리지 및 저장 데이터의 무결성 보장 기술 및 저장 자원 또는 데이터로의 접근을 제어하기 위한 사용자 인증, 인가 및 접근 제어 기술 등이 포함된다. 제공자 측면에서 적용 가능한 클라우드 스토리지 보안 기술은 클라우드 서비스의 제공 형태에 따라 다시 인프라(IaaS), 플랫폼(PaaS), 서비스(SaaS) 영역에서의 스토리지 보안 기술로 구분할 수 있다.

사용자 단말 보안 측면에서는 NIST(National Institute of Standards and Technology)가 사용자 단말 스토리지 암호화, 암호 키 관리 및 사용자 인증에 대한 가이드라인 문서[16]를 발간하여 주요 주제로 단말 무결성과 데이터 접근 제어 등을 다루었다. 단말에 저장된 플랫폼의 무결성을 보장하기 위해 단말의 안전한 부팅[56]을 지원하는 기능을 제공하기도 하며 단말에서의 데이터 유출 방지를 위한 ARM TrustZone을 이용한 모바일 클라우드 스토리지 서비스에서 데이터 유출 방지 연구[40]와 사용자 단말에서의 접근 제어 기술[48] 등은 ARM TrustZone과 같은 하드웨어 지원 기능을 활용하여 플랫폼 무결성 검증 및 파일 암복호화 기능을 안전하게 제공한다. 이외에도 저장된 데이터의 완전 삭제 기술[59] 및 저장된 데이터로의 접근을 포함하는 데이터의 이동 흐름에 대한 모니터링 및 기록을 위해 정보 흐름 추적 기술(Information Flow Tracking)[26,27]과 같은 데이터 흐름 모니터링 기술들이 데이터에 대한 접근 제어, 로깅 및 감사 추적을 위한 용도로 연구되고 있다.

인프라 형태의 클라우드 서비스(IaaS) 제공자에서의 스토리지 보안 기술은 주로 기반 보호 기술로써 자료유출 방지 및 스토리지 암호화 기술, 안전한 데이터 중복 제거 등의 연구가 활발하게 이루어지고 있다. 자료 유출 방지(DLP) 기술은 기밀 정보에 대한 허가받지 않은 접근, 사용, 전송을 막기 위한 기술 또는 심층 컨텐츠 분석을 통해 사용, 전송중이거나 변경 없는 데이터를 중앙 정책에 기반하여 확인, 모니터링, 보호하는 기술로 정의되며[49] 산업계에서도 많은 제품이 출시되어 있다. 최근 학계에서는 스토리지 수준 IDS(Intrusion Detection System)[34,35,36], 데이터 labeling, 데이터 필터[28] 등의 연구들이 이루어지고 있다.

스토리지 데이터의 암호화를 위해 IEEE P1619[31]는 블록 기반 스토리지 장치에서의 데이터 암호화 보호에 대한 표준을 제시하였으며 데이터 유실 및 유출과 관련된 식별, 규정, 접근 제어 및 관련 속성과 관련해서는 SAML, X.509 인증 등이 사용되고 있다[30]. 일반적인 스토리지 암호화는 전체 디스크 암호화[32,33], 가상 디스크 암호화 및 볼륨 암호화[47], 그리고 일반적으로 가장 널리 사용되는 파일/폴더 암호화등이 있다. 이외에도 다양한 클라우드용 경량 암호화 기술[52,53]이 연구되고 있으며 산업계에서도 구글은 2013년부터 public cloud 서비스의 스토리지에 저장되는 모든 데이터를 자동 암호화[5]하고, 이외에도 클라우드 서비스 제공자들은 사용자 데이터의 암호화 기능을 기본/옵션으로 제공하고 있다.

중복 제거(Deduplication) 기술은 클라우드 저장 공간 내 중복된 데이터에 대한 저장 공간 절감 및 통신 부담을 덜기 위해 적용되나 보안에

대한 고려가 없을 경우 side-channel attack[49]에 의해 데이터의 저장 여부를 공격자에게 노출시키거나 암호화에 의해 중복 제거가 의미없게 되는[50] 문제가 있다. 최근에는 사용자 단말[24], 클라우드 서비스 제공자[21,22,25,50] 측면에서 보안성을 고려한 저장된 데이터에 대한 중복 제거 기술들이 연구되고 있다.

이외에도 스토리지 무결성 보장과 관련하여 클라우드 환경에서 사용되는 데이터 무결성 검증[37,38,71] 기법 등이 연구되고 있으며 감사 기법[41,42,43,44]들도 제안되고 있다.

플랫폼 형태의 서비스 제공자 측면에서의 스토리지 보안 기술은 분산 데이터의 접근 제어 및 사용자 데이터 보호와 관련된 연구들이 이어지고 있다. 특히 클라우드 스토리지에 저장된 대용량 정보를 암호화된 상태로 저장하면서 검색 등 필요한 작업을 수행할 수 있도록 하는 검색 가능 암호[45,46]가 빅데이터 분야를 포함하여 연구되고 있다.

클라우드 서비스 제공자 측면에서의 스토리지 보안 기술은 사용자 데이터 보호가 주를 이루며 데이터 권한 관리 및 기밀성 보장을 위한 데이터 암호화[54,55] 등이 연구되고 있다.

또한 서비스 유형에 관계없이 저장된 데이터의 복구를 용이하게 하기 위한 기술[57,67], 및 정상적으로 삭제되거나 가입 해지된 사용자의 데이터에 대한 클라우드 스토리지로부터의 완전 삭제 기술이 저장된 데이터의 복구를 어렵게 하는 의미로 통용되어 클라우드 안티 포렌식 기술[69]과 연계되어 연구되고 있다. 그리고 스토리지 저장 데이터로의 입출력과 관계된 보안 감사 및 이벤트 관리/모니터링과 관련하여 스토리지 저장 데이터로의 입출력에 대한 모니터링[18,19,20] 기법들도 연구되고 있다.

클라우드 환경에서의 보안 사고에 대한 대응 및 조사와 관련하여 디지털 포렌식을 지원할 수 있는 스토리지 관련 기술 또한 중요한 이슈로 부각되고 있다. 이와 관련하여 [60,61,64,66]에서는 클라우드 환경에서의 포렌식 조사를 위한 기술적 이슈들을 다루었으며 로깅 기반 포렌식 기술[62,63,68] 및 증거 추적 및 재구성을 위한 포렌식 프레임워크[65], 클라우드 환경에서의 포렌식 기술[70] 등이 제안되었다.

4. 결 론

클라우드 컴퓨팅 환경은 증가하는 데이터의 양과 늘어나는 사용자 수에 따라 다양한 보안 위협에 노출되고 있으며 여러 구성 요소들 중에서도 클라우드 스토리지 시스템은 사용자의 데이터 유출, 유실, 악의적인 서비스 제공자로부터의 공격, 그리고 클라우드 효율성을 위한 공유 기술에서 파생되는 문제와 같은 데이터의 중앙집중화에 따른 보안 위협을 받을 수 있기 때문에 보안상의 대비가 반드시 이루어져야 한다.

본 고에서는 기존에 분류되었던 클라우드 스토리지와 관련된 보안 위협의 분류들을 갱신하고, 해당 보안 위협과 관련된 연구 동향을 알아보았다. 이후, 이를 바탕으로 클라우드 스토리지 시스템의 보안 위협에 대응하는 연구를 진행할 계획이다.

참 고 문 헌

[1] CSA Top Threats Working Group, “The Treacherous 12: Cloud Computing Top Threats in 2016,” Cloud Security Alliance (CSA), Feb 2016.

[2] 클라우드컴퓨팅 연구조합, 차세대컴퓨팅학회, “2015 클라우드 컴퓨팅 기술 스택,” 2016.01
[3] 정성재, 배유미, “클라우드 보안 위협요소와 기술 동향 분석,“ 보안공학연구논문지, 2013.04
[4] Google Compute Engine Incident #15056, https://status.cloud.google.com/incident/compute/15056#5719570367119360, (2016.07.06)
[5] Dave Barth, “Google Cloud Storage now provides server-side encruption,” https://cloudplatform.googleblog.com/2013/08/google-cloud-storage-now-provides.html, Google Cloud Platform Blog (2016.07.06)
[6] 주정호 외, “클라우드 컴퓨팅 보안 위협에 대처한 저장장치 가상화 시스템 보안 요구 사항 제안,” 보안공학연구논문지 Vol.11, No.6, 2014
[7] ITU-T Study Group 17, “Recommendation ITU-T X.1601 Security framework for cloud computing,” 2015.10, available on http://handle.itu.int/11.1002/1000/12613
[8] ENISA, “ENISA threat Landscape 2015 (ETL 2015),“ 2016.01, available on https://www.enisa.europa.eu/publications/etl2015
[9] 김홍연 외, "GLORY-FS: 대규모 인터넷 서비스를 위한 분산 파일 시스템." The Journal of The Korean Institute of Communication Sciences, 30.4 (2013.3): 16-22.
[10] Hadoop Project, http://hadoop.apache.org
[11] K. Shvachko et al., “The Hadoop Distributed File System.” In Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST) (MSST '10). IEEE Computer Society, Washington, DC, USA, 1-10.
[12] OpenStack Swift, http://www.openstack.org
[13] 김진수, 김태웅, “OwFS: 대규모 인터넷 서비스를 위한 분산 파일 시스템,” 정보과학회지 27권 5호, 2009.05.
[14] GlusterFS, “Open source, distributed file system,” http://www.gluster.org, March 2013.
[15] IETF, RFC5661, Network File System (NFS) Version 4 Minor Version 1 Protocol, Jan. 2010
[16] K Scarfone et al., “Guide to Storage Encryption Technologies for End User Devices,” NIST Special Publication 800-111, Nov. 2007.
[17] 박정수, 배유미, 정성재. (2013.5). 클라우드 컴퓨팅을 위한 클라우드 스토리지 기술 분석. 한국정보통신학회논문지, 17(5), 1129-1137.
[18] Xuxian Jiang et al., “Stealthy malware detection and monitoring through VMM-based “out-of-the-box” semantic view reconstruction,“ ACM Trans. Inf. Syst. Secur.13, 2, Article 12 (March 2010), 28 pages.
[19] C. Benninger et al., “Maitland: Lighter-weight VM introspection to support cyber-security in the cloud,” CLOUD’12, pages 471-478, June 2012.
[20] Wolfgang Richter, “Agentless cloud-wide monitoring of virtual disk state,” In Proceedings of the 2014 workshop on PhD forum (PhD forum '14). ACM, New York, NY, USA, 15-16.
[21] W. Richter et al., “Agentless Cloud-Wide Streaming of Guest File System Updates,” IC2’14, IEEE Computer Society, Washington, DC, USA.
[22] P. Puzio et al., “ClouDedup: Secure Deduplication with Encrypted Data for Cloud Storage.” 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, Bristol, 2013, pp. 363-370.
[23] J. Stanek et al., "A secure data deduplication scheme for cloud storage." International Conference on Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2014.
[24] N. Kaaniche and M. Laurent, "A Secure Client Side Deduplication Scheme in Cloud

Storage Environments," 2014 6th International Conference on New Technologies, Mobility and Security (NTMS), Dubai, 2014, pp. 1-7.
[25] Mark W. Storer et al., "Secure data deduplication," In Proceedings of the 4th ACM international workshop on Storage security and survivability (StorageSS '08). ACM, New York, NY, USA, 1-10.
[26] Man-Ki Yoon et al., "PIFT: Predictive Information-Flow Tracking," In Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '16). ACM, New York, NY, USA, 713-725.
[27] C. Wang and Shiuhpyng Winston Shieh. "SWIFT: Decoupled System-Wide Information Flow Tracking and its Optimizations." Journal of Information Science and Engineering 31.4 (2015): 1413-1429.
[28] Christos Gkantsidis et al., "Rhea: automatic filtering for unstructured cloud storage," In Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation (nsdi'13), USENIX Association, Berkeley, CA, USA, 343-356.
[29] Dongyoung Koo et al., "Secure and efficient data retrieval over encrypted data using attribute-based encryption in cloud storage," Comput. Electr. Eng. 39, 1 (January 2013), 34-46.
[30] 허의남, "퍼스널 클라우드 보안 기술과 프라이버시," TTA 저널 no.139, 2012, 65-69.
[31] IEEE Security in Storage Working Group, "IEEE P1619/D16 Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices," IEEE Computer Society Committee, May 2007.
[32] E. Casey and Gerasimos J. Stellatos, "The impact of full disk encryption on digital forensics." ACM SIGOPS Operating Systems Review42.3 (2008): 93-98.
[33] E. Casey et al. "The growing impact of full disk encryption on digital forensics," Digital Investigation 8.2 (2011): 129-134.
[34] A. Bacs et al., "Slick: an intrusion detection system for virtualized storage devices," In Proceedings of the 31st Annual ACM Symposium on Applied Computing (SAC '16). ACM, New York, NY, USA, 2033-2040.
[35] Adam G. Pennington et al., "Storage-Based Intrusion Detection," ACM Trans. Inf. Syst. Secur. 13, 4, Article 30 (December 2010), 27 pages.
[36] Anjo Vahldiek-Oberwagner et al., "Guardat: enforcing data policies at the storage layer," In Proceedings of the Tenth European Conference on Computer Systems (EuroSys '15). ACM, New York, NY, USA, , Article 13, 16 pages.
[37] 은하수 외, "Outsourced Storage Auditing Scheme using Coefficient Matrix." 정보처리학회논문지 2.11 (2013): 483-488.
[38] 김태연, 조기환, "클라우드 컴퓨팅에서 동적 데이터 무결성을 위한 개선된 감사 시스템," 한국정보통신학회논문지 Vol.19, No.8, (2015.08): 1818-1824.
[39] Amazon S3, "Protecting Data Using Encryption," http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html (2016.07.06)
[40] 신재복 외, "클라우드 스토리지 서비스를 위한 ARM TrustZone 기반의 안전한 데이터 관리 기법," 대한임베디드공학회 추계학술대회, 2012. 11. 85-88.
[41] 백목련 외, "안전한 클라우드 스토리지를 위한 프라이버시 보장 자체 인증 공공 감사," 정보과학회논문지, 43(4), 497-508.
[42] 권오민, 윤현수, "클라우드 데이터 무결성 검증 방법 조사," 한국정보과학회 학술발표논문집, 2013.6, 689-691.
[43] 신수연, 권태경, "손상 클라우드 식별 가능한 다

중 클라우드 일괄 감사 기법에 관한 연구," 정보보호학회논문지 25(1), 2015.2, 75-82.
[44] C. Wang et al., "Privacy-Preserving Public Auditing for Secure Cloud Storage," in IEEE Transactions on Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.
[45] M. Bakhtiari et al., "Secure Search Over Encrypted Data in Cloud Computing," Advanced Computer Science Applications and Technologies (ACSAT), 2013 International Conference on, Kuching, 2013, pp. 290-295.
[46] C. Liu et al., "Efficient Searchable Symmetric Encryption for Storing Multiple Source Data on Cloud," Trustcom/BigDataSE/ISPA, 2015 IEEE, Helsinki, 2015, pp. 451-458.
[47] K. Beer and R. Holland, "Securing Data at Rest with Encryption," Amazon Web Services, Nov. 2013.
[48] 신재복 외, "모바일 클라우드 스토리지 서비스에서의 데이터 보안을 위한 데이터 접근 제어 및 보안 키 관리 기법," 대한임베디드공학회논문지 8(6), 303-309.
[49] S. Alneyadi et al., "A survey on data leakage prevention systems," Journal of Network and Computer Applications 62, 2016, 137-152.
[50] D. Harnik et al, "Side channels in cloud services: Deduplication in cloud storage." IEEE S&P, 8(6) 2010, 40-47
[51] S. Keelveedhi et al., "DupLESS: server-aided encryption for deduplicated storage." 22nd USENIX Security Symposium (USENIX Security 13). 2013.
[52] S. Matsuda and S. Moriai, "Lightweight Cryptography for the Cloud: Exploit the Power of Bitslice Implementation," Cryptographic Hardware and Embedded Systems (CHES), 2012, 408-425
[53] S. Belguith et al., "Enhancing Data Security in Cloud Computing Using a Lightweight Cryptographic Algorithm," Eleventh International Conference on Autonomic and Autonomous Systems, 2015.
[54] Y. Chou et al., "Enforcing confidentiality in a SaaS cloud environment," Telecommunications Forum (TELFOR), 2011, 90-93.
[55] S. Rehman and R. Gautam, "Research on ACcess Control Techniques in SaaS of Cloud Computing," Security in Computing and Communications, Springer, 2014, 92-100.
[56] R. Wilkins and B. Richardson, "UEFI Secure Boot in Modern Computer Security Solutions," UEFI Forum, Sep. 2013.
[57] O. Khan et al., "Rethinking Erasure Codes for Cloud File Systems: Minimizing I/O for Recovery and Degraded Reads," USENIX FAST, 2012.
[58] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," 2011, http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf
[59] Xingjie Yu et al., "Remotely wiping sensitive data on stolen smartphones," In Proceedings of the 9th ACM symposium on Information, computer and communications security (ASIA CCS '14). ACM, New York, NY, USA, 537-542.
[60] D. Birk and C. Wegener, "Technical Issues of Forensic Investigations in Cloud Computing Environments," Systematic Approaches to Digital Forensic Engineering (SADFE), 2011 IEEE Sixth International Workshop on, Oakland, CA, 2011, pp. 1-10.
[61] S. Alqahtany et al., "Cloud Forensics: A Review of Challenges, Solutions and Open Problems," Cloud Computing (ICCC), 2015 International Conference on, Riyadh, 2015, pp. 1-9.
[62] S. Zawoad et al., "Towards Building Forensics Enabled Cloud Through Secure Logging-as-a-Service," in IEEE Transactions

on Dependable and Secure Computing, vol. 13, no. 2, pp. 148-162, March-April 1 2016.
[63] T. Sang, "A Log Based Approach to Make Digital Forensics Easier on Cloud Computing," Intelligent System Design and Engineering Applications (ISDEA), 2013 Third International Conference on, Hong Kong, 2013, pp. 91-94.
[64] A. K. Mishra et al., "Cloud Forensics: State-of-the-Art and Research Challenges," Cloud and Services Computing (ISCOS), 2012 International Symposium on, Mangalore, 2012, pp. 164-170.
[65] S. Almulla et al., "A Distributed Snapshot Framework for Digital Forensics Evidence Extraction and Event Reconstruction from Cloud Environment," 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, Bristol, 2013, pp. 699-704.
[66] V. M. Katilu et al., "Challenges of Data Provenance for Cloud Forensic Investigations," Availability, Reliability and Security (ARES), 2015 10th International Conference on, Toulouse, 2015, pp. 312-317.
[67] 남궁재웅 외, “포렌식 관점의 파티션 복구 기법에 관한 연구,” Journal of The Korea Institute of Information Security & Cryptology, 23(4), 2013.08, 655-665.
[68] S. Khan et al., “Cloud Log Forensics: Foundations, State of the Art, and Future Directions,” ACM Computing Syrveys 49(1), Feb. 2016.
[69] NIST Cloud Computing Forensic Science Working Group IT Laboratory, “Draft NISTIR 8006, NIST Cloud Computing Forensic Science Challenges,” NIST, 2014.
[70] 정일훈 외, “IaaS 유형의 클라우드 컴퓨팅 서비스에 대한 디지털 포렌식 연구,“ 정보보호학회논문지 21(6), 2011.12, 55-65.
[71] 김등화 외, “사설 클라우드 환경에서 수집된 VM 데이터의 무결성 입증과 관련 포렌식 도구의 신뢰성 검증,” 정보보호학회논문지 23(2), 2013.4, 223-230.

저 자 약 력

황 우 민

이메일 : wmhwang@nsr.re.kr

- 2015년 한국과학기술원 전기및전자공학과 (박사)
- 2014년~현재 국가보안기술연구소 연구원
- 관심분야 : 클라우드, 고성능 컴퓨팅, 시스템 보안, 가상화 보안

김 성 진

이메일 : ksj1230@nsr.re.kr

- 2010년 포항공과대학교 정보통신학 (석사)
- 2010년~현재 국가보안기술연구소 선임연구원
- 관심분야 : 컴퓨터 보안, 악성코드 분석

김 형 천

이메일 : khche@nsr.re.kr

- 2011년 고려대학교 정보보호대학원 (박사)
- 2001년~현재 국가보안기술연구소 책임연구원
- 관심분야 : 클라우드 보안, 컴퓨터 보안

특집 05

입출력 패턴에 기반한 스토리지 서버의 소비전력 절감 가능성 및 성능 평가 방법

박찬영 · 이재면 · 강경태 (한양대학교)

목 차

1. 서 론

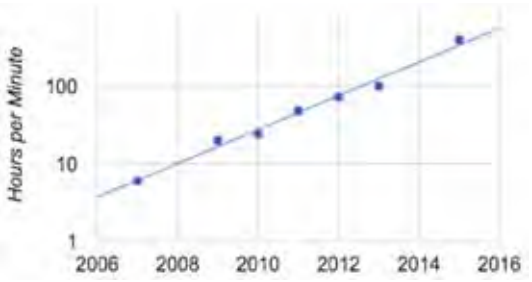
사용자가 생산하는 데이터의 양이 급격하게 증가하며 이에 대응하기 위한 서비스 제공자의 고민이 늘고 있다. 그림 1과 같이 Youtube 서비스 이용자들은 1분에 400시간의 동영상을 올리며, 이를 위해 매일 1페타바이트(1,000,000GB)의 새로운 스토리지가 필요하다[1]. 사용자는 업로드된 자신의 자료가 언제까지나 보관되기를 원하지만, 이는 고스란히 서비스 제공자에게 큰 부담으로 작용하게 된다. 데이터 저장을 위해 데이터센터의 스토리지 서버를 끊임없이 증축해야 하며, 빠른 서비스를 위해 스토리지 서버를 활성 상태로 유지해야 하기 때문이다[2]. 이에 따라 데이터센터가 소비하는 전력은 지속해서 증가하게 되며, 데이터센터의 소비전력 절감이 매우 중요한 문제로 대두되고 있다.

이 문제를 데이터의 접근 패턴 관점에서 해결하려는 방안이 있다. 이는 나이[1]가 많은 데이터일수록 접근율이 낮다는 아이디어에 기초한다. 데이터의 나이와 접근율의 상관관계를 놓고 그래프를 그려보면, 나이가 적은 데이터가 서버 전체 접근율의 대부분을 차지하게 되는 tail latency 형태의 그래프가 그려진다[3]. 이때 접근 요청이 거의 없는 데이터를 콜드 데이터, 접근 요청이 빈번한 데이터를 핫 데이터라고 부르며, 콜드 데이터만을 모아 적절한 정책으로 관리하여 데이터센터의 소비전력을 절감할 수 있다.

이와 같이, 현재 산업계는 콜드 데이터에 집중하여 데이터센터의 소비전력을 절감하고자 한다[4,5]. Open Compute Project[2] (OCP)에서는 콜드 데이터를 저장하기 위한 저전력 스토리지 서버(콜드 스토리지)의 명세서를 공개하였다[6]. Facebook은 콜드 스토리지를 디스크와 블루레이

1) 생성 시각으로부터 오래된 데이터인지 아닌지를 나이가 많다, 적다는 것으로 표현한다.

2) Open Compute Project는 하드웨어를 재구성하여 더욱 효율적이고, 유연하며, 확장성 있게 만드는 데 초점을 둔 컴퓨팅 인프라 구조 협력 커뮤니티이다[6].



(그림 1) Youtube upload rate[1]

로 확장하고 있으며, 앞으로 플래시 메모리를 기반으로 한 콜드 스토리지를 내놓을 계획이라고 발표하였다[7]. 또한, Google은 2016년 USENIX가 개최하는 국제 학술대회에서 발표된 기고문을 통해 콜드 데이터를 위한 디스크 및 관리 도구의 필요성을 주장한 바 있다[1].

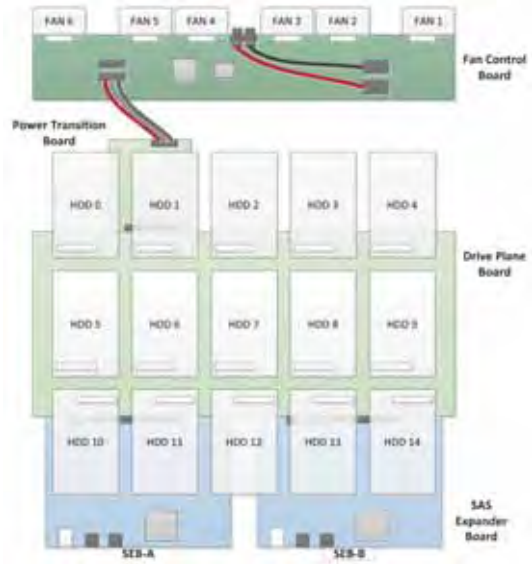
이처럼 스토리지 서버의 소비전력 절감을 위해 다양한 연구가 진행되고 있다. 하지만 소비전력을 절감하였다 하더라도 서비스가 요구하는 수준의 응답시간을 보장하지 못한다면, 연구 결과를 실제로 적용하는데 한계가 있다. 따라서 서버의 성능 평가 방법 역시 중요하다. 하지만 대규모 인프라에 초점을 맞춘 콜드 스토리지 연구는 학계에서 성능 평가를 진행하기에 규모 면에서 어려움이 있다. 이를 해결하기 위해 다양한 워크로드를 생성해낼 수 있는 벤치마킹 도구로 본 논문에서는 Yahoo! Cloud Serving Benchmark (YCSB)를 소개한다.

2. 데이터센터의 소비전력

데이터센터가 소모하는 전력은 어느 정도인가? 미국의 경우 이미 수년 전부터 데이터센터가 소비하는 전력이 뉴욕 시가 소모하는 전력의 2배를 넘어섰다[8]. 국내 데이터센터의 경우 2013년 기준 약 26억kWh의 전력을 소모하였다. 이는 한 달 동안 약 1,200만 가구가 사용하는 전력량과 비슷하다[9]. 또한, 새로운 데이터가 생성될수록 데이터센터가 소모하는 전력량은 늘어날 수밖에 없으며, 그 상승세 역시 가파른 실정이다.

이처럼 많은 양의 전력을 소모하는 데이터센터는 기업의 골칫거리이다. 우선 값비싼 전기세로 인해 유지비용이 비싸고, 친환경 에너지로만 전력을 공급받지 않는 이상 환경오염의 질책을 피할 수 없다[8-9]. 이를 해결하기 위해 다양한 산업계의 노력이 있었으며, 대표적으로 OCP의 콜드 스토리지가 있다. 콜드 스토리지는 에너지 절감을 위해, 연산 능력을 손해 보더라도 낮은 가격과 저전력으로 요청이 적은 콜드 데이터를 처리하는 스토리지 서버다.

OCP의 콜드 스토리지는 Open Vault[3]의 기준(그림 2)을 따르며 소비 전력 최소화를 위해 몇 가지 요소를 변경하였다. 총 6개의 fan 모듈에서 2개를 제거하여 4개의 fan 모듈이 있다. 그리고 compute node의 사양을 낮추었으며 하나의 스토리지 서버



(그림 2) Open Vault's major system components[6]

3) OCP에서 정의한 스토리지 서버를 모두 아울러 Open Vault라 한다.

에 하드디스크 2개만 동작하도록 하여 총 하드디스크의 개수와 상관없이 2개 분량의 전력을 소비하도록 설계되었다. 실제로 Facebook은 콜드 스토리지를 적용하여, 이전 대비 38% 이상의 에너지 효율과 24% 이상의 가격 경쟁력을 얻은 바 있다.

또한, 지속적인 플래시 가격의 하락으로 인하여 플래시를 이용한 콜드 스토리지 연구가 진행되고 있다. 플래시는 하드디스크보다 크기는 작고, 소비전력이 낮으며 진동과 발열이 적은 장점이 있다. 따라서 플래시로 콜드 스토리지를 구성하게 되면, 높은 집적도와 낮은 소비전력을 기대할 수 있다. 하지만 아직 하드디스크에 비해 4배 높은 가격으로 인해 가까운 시일 내 상용화는 어려울 것으로 예상한다.

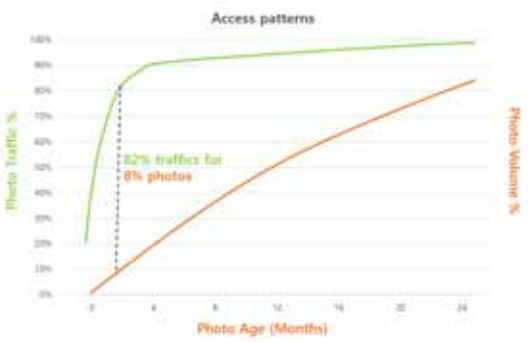
데이터센터의 막대한 소비전력 문제는 관련 업계에 종사하는 사람뿐만 아니라 현대사회의 큰 주목을 받고 있으며[8-9], 이를 해결하기 위한 저전력 서버 연구가 산업계를 중심으로 활발하다. 특히, OCP를 중심으로 하드웨어 연구가 진행되고 있을 뿐만 아니라, 콜드 스토리지에서 최적의 성능을 내기 위한 소프트웨어 연구도 진행되고 있다. 소프트웨어 연구는 콜드 데이터의 분류로부터 출발하며, 이를 위해 데이터 접근 패턴의 분석이 선행되어야 한다[10,11].

3. 데이터 접근 패턴 분석

데이터 접근 패턴 분석을 위해 데이터의 정렬이 필요하다. 이때 데이터의 나이를 정렬 기준으로 삼는 것이 일반적이다. 특히, 콜드 스토리지 운영 시 좋은 효과를 기대할 수 있는 소셜 네트워크, 동영상 공유 서비스와 같은 대규모 서비스의 경우 나이가 적은 데이터가 전체 트래픽의 대부분을 차지하는 형태의 패턴이 뚜렷하였다. 관련 사례로 Facebook과 LINE 서비스의 데이터 접근 패턴을 분석하였다.

3.1 Facebook의 데이터 접근 패턴

그림 3은 유명 소셜 네트워킹 서비스 제공사인 Facebook이 공개한 데이터 접근 패턴이다. 업로드된 사진 데이터를 나이로 정렬한 후 전체 트래픽에서 차지하는 비율을 나타낸 결과, 어린 나이의 데이터가 차지하는 트래픽 비율이 극단적으로 높았다. 이들이 공개한 수치에 의하면 2013년 기준 약 8% 사진 데이터가 전체 트래픽의 82%를 차지하였다.

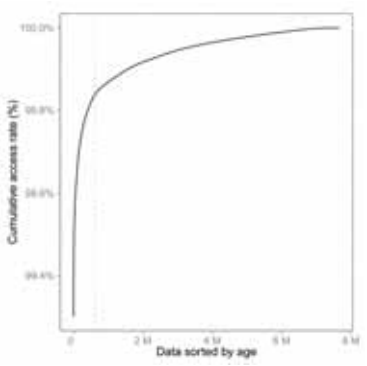


(그림 3) Facebook data access patterns, 2013[2]

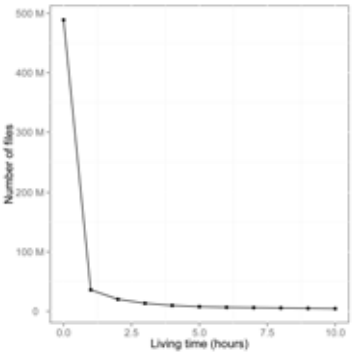
3.2 LINE의 데이터 접근 패턴

LINE은 무료 통화 및 메신저 애플리케이션이다. 사용자는 LINE을 이용하여 사진을 교환할 수 있다. 그림 3은 LINE에서 교환되는 사진 데이터를 나이 순으로 정렬하여 데이터 접근 패턴을 분석한 결과이다. 나이가 어린 8%의 데이터가 99.8%의 트래픽을 점유하였으며 앞선 Facebook과 유사한 형태를 보였다[12].

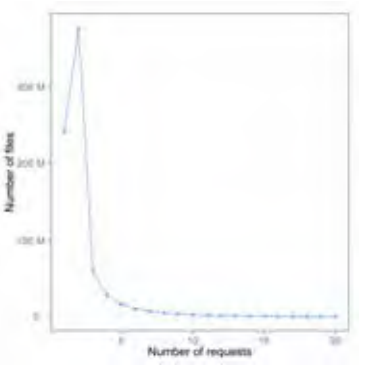
더욱 정확한 콜드 데이터 분류 정책을 위해 다양한 기준에서 데이터를 분석할 필요가 있다. 그림 5는 데이터의 생성 시각부터 마지막 요청 시각까지의 시차를 기준으로, 그림 6은 데이터의 요청 횟수를 기준으로 하여 데이터를 분석한 결과이다. 전자



(그림 4) LINE 데이터 접근 패턴[12]



(그림 5) LINE 이미지 파일 분포 - 활동시간 기준[12]



(그림 6) LINE 이미지 파일 분포 - 요청횟수 기준[12]

의 경우 데이터 생성 시각과 마지막 요청 시각의 차이(활동시간)가 4시간 이하인 데이터가 전체 데이터의 약 64%를 차지하였으며, 약 4%의 데이터가 활동시간 1시간에서 2시간 사이에 존재하였다. 이는 곧 활동시간 1시간 이후의 데이터는 접근 요청이 거의 없으며 콜드 데이터로 분류 가능함을 의미한다. 후자의 경우는 31.62%의 데이터가 한 번의 요청을 받고, 49.24%의 데이터가 두 번의 요청을 받는다. 따라서 약 80%의 데이터가 2번 이하의 요청 횟수를 가지며, 이는 일대일 대화방이 많은 LINE 서비스의 특징을 반영한 결과이다. 한번 접근된 사진 데이터는 장치에 저장되므로 추후 접근 시 재다운로드할 필요가 없다. 따라서 대화방의 인원수만큼 파일이 접근되면 콜드 데이터로 분류가 가능하다[12].

위 결과를 통해 Facebook과 LINE 모두 소수의 어린 데이터가 트래픽의 대부분을 차지하고 콜드 데이터가 많은 형태임을 확인하였다. 또한, LINE 서비스의 경우 더욱 정확한 콜드 데이터 선정을 위해 데이터의 활동시간, 요청횟수를 기준으로 파일의 분포를 분석하였으며, 다양한 분석 결과를 복합적으로 적용하여 콜드 데이터 분류가 가능했다. 분류된 데이터를 콜드 스토리지에 저장 및 관리한다면, 데이터센터의 소비전력을 절감할 수 있다.

하지만 모든 서비스가 위와 같은 데이터 접근 패턴을 보이는 것은 아니다. Facebook, LINE과 같은 소셜 네트워크 서비스는 프론트 엔드부터 시간을 기준으로 한 콜드 데이터 분류에 유리하도록 설계되어 있다. 사용자에게 보이는 데이터는 기본적으로 시간이 지날수록 스크롤 아래로 밀려나게끔 설계되어있으며, 이러한 특성으로 인해 데이터 접근 패턴의 시간 의존성이 높아진다. 반면에 검색 엔진과 같이 시간 의존성이 높지 않은 서비스는 데이터 접근 패턴의 분석과 콜드 데이터 분류의 기준을 달리할 필요가 있다. 이처럼 다양한 서비스에 적합한 콜드 데이터 선정 기준은 콜드 스토리지의 적용 분야를 확장함에 있어 도전 과제로 남아있다.

4. 서버의 성능 평가

서버의 에너지 효율을 높이는 연구를 함에 있어서도, 서비스 운영을 위해 일정 수준의 응답시간 보장이 필요하다. 콜드 스토리지는 그 특성상 데이터센터를 소유하고 대규모 서비스를 운영하는 곳에서 필요로 한다. 따라서 대규모 워크로드에서 서버의 성능을 쉽고 공정하게 비교할 방법이 필요하다.

하지만, 데이터센터를 보유하기 힘든 학계는 콜드 스토리지 실험을 수행하는 데 한계가 있다. 또한,

〈표 1〉 YCSB 핵심 패키지의 워크로드[14]

Workload	Operations	Record selection	Application example
A-Update heavy	Read: 50% Update: 50%	Zipfian	Session store recording recent actions in a user session
B-Read heavy	Read: 95% Update: 5%	Zipfian	Photo tagging; add a tag is an update, but most operations are to read tags
C-Read only	Read: 100%	Zipfian	User profile cache, where profiles are constructed elsewhere (e.g., Hadoop)
D-Read latest	Read: 95% Insert: 5%	Latest	User status updates; people want to read the latest statuses
E-Short ranges	Scan: 95% Insert: 5%	Zipfian/Uniform	Threaded conversations, where each scan is for the post in a given thread (assumed to be clustered by thread id)

실 서비스 워크로드가 없어 실험의 신뢰성을 보장하기 어렵다. 이에 벤치마킹 툴을 개발하여 워크로드 규모를 축소하거나 가상의 워크로드로 시뮬레이션하고 있다[13]. 이 접근 방법은 다양한 실험을 할 수 있는 장점이 있지만, 실제 서비스와의 유사성 문제를 안고 있다.

스토리지 서버를 실험하는 다양한 벤치마킹 툴이 존재한다[14-17]. 그 중 YCSB는 온라인 클라우드 서비스를 제공하는 데이터베이스 벤치마킹 툴이다[14]. 클라우드 서비스의 요청 패턴과 유사하도록 다중 클라이언트 기능을 제공하고, 워크로드를 사용자 요구에 맞게 수정할 수 있다. 특히, 요청 패턴을 5가지 정책(Constant, uniform, zipfian distribution, sequential, hotspot)에 따라 생성함으로써, 실제 서비스 요청과 유사한 패턴을 보일 수 있다.

오픈소스인 YCSB는 NoSQL 데이터베이스 전문 벤치마킹 툴이지만, Key-value Store 기반의 오브젝트 스토리지 기반 파일시스템도 연결 객체(binding)를 개발하여 실험 가능하다[18]. 현재 Ceph의 오브젝트 스토리지 RADOS binding을 공식 지원하고 있다[19].

사용자는 YCSB에 CRUD+S(읽기, 쓰기, 업데이트, 삭제, 스캔) 동작의 비중을 달리하여 다양한 요청 패턴을 실험할 수 있으며, 표 1은 YCSB에서 제공하는 대표 워크로드이다. 워크로드 별로 초당 명령 처리속도와 평균 대기시간 등 서버 성능 평가에 필요한 지표를 얻고 성능을 비교할 수 있다.

5. 결 론

데이터센터의 소비전력 절감이 TCO 관점의 문제를 넘어 매우 중요한 사회적 문제로 떠오르고 있다. 그린피스를 비롯한 환경단체로부터 데이터센터의 많은 전력 소비를 지적받고 있으며 세계 각국의 정부 부처는 데이터센터의 에너지 규제에 관심을 보이고 있다[20,21].

데이터센터의 소비전력 절감 가능성을 확인하기 위해 애플리케이션의 데이터 접근 패턴을 분석하였다. 그 결과 특정 서비스에서 나이가 어린 소수의 데이터가 트래픽의 대부분을 차지하였으며, 콜드 데이터에 집중한 소비전력 절감 연구가 진행되고 있음을 확인하였다. 또한, 더욱 정확한 콜드 데이터 선정을 위하여, 서비스별 특성을 고려한 데이터 접근 패턴 분석의 기준이 필요함을 확인하였다.

콜드 스토리지의 명세를 공개한 OCP를 중심으로 하드웨어 연구가 활발히 진행되고 있으며, 오픈

소스 분산파일시스템 커뮤니티를 중심으로 소비전력 절감에 초점을 맞춘 연구 성과물들이 점차 발표되고 있다. 하지만 스토리지 서버 연구는 실제 서비스 규모가 방대하여 학계에서 연구하기 한계가 있다. 이를 위해 YCSB와 같은 벤치마킹 도구가 있으며, 스토리지 서버 연구의 성능 지표 검증이 가능함을 확인하였다. 그 결과, 학계에서도 관련 연구가 더욱 활성화될 것을 기대한다.

참 고 문 헌

[1] Google, Disks for Data Centers, 2016.

[2] OSCON – The OCP keynote by Jay Parikh. [Online]. Available: http://www.oscon.com/oscon2013/public/schedule/detail/29558

[3] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, “Analyzing the Video Popularity Characteristics of Large-Scale User Generated Content Systems,” IEEE/ACM Transactions on Networking (TON), Vol. 17, no. 5, pp. 1357-1370, 2009.

[4] E. Thereska, A. Donnelly, and D. Narayanan, “Sierra: Practical power-proportionality for data center storage,” Proc. the 6th Conf. Computer Systems (Eurosys 11), pp. 169-182, 2011.

[5] C. Albrecht, A. Merchant, M. Stokely, M. Waliji, F. Labelle, N. Coehlo, X. Shi, and E. Schrock, “Janus: Optimal flash provisioning for cloud storage workloads,” Proc. USENIX Ann. Technical Conference (ATC 13), pp. 91-102, 2013.

[6] Open Compute Project. [Online]. Available: http://www.opencompute.org

[7] Open Compute Project US Summit 2016 by Jay Parikh. [Online]. Available: http://www.opencompute.org/wiki/Main_Page/Summit/2016US

[8] NRDC, Data Center Efficiency Assessment, 2014.

[9] 그린피스, “당신의 인터넷은 깨끗한가요?,” 2015.

[10] 이재면, 강경태, “OCP Cold Storage 테스트베드,” 정보과학회 컴퓨팅의 실제 논문지, 제22권, 제3호, pp.151-156, 2016.

[11] 이재면, 정해건, 원지웅, 강경태, “OCP Cold Storage 정책 평가를 위한 테스트 베드,” 한국컴퓨터종합학술대회(KCC), 2015.

[12] J. Lee, C. Song, and K. Kang, “Energy-efficient Storage Policy for Instant Messenger Services,” in Proc. 5th IEEE International Conference on Big Data and Cloud Computing (BDCloud), pp. 38-44, 2015.

[13] Y. Wang, M. Kapritsos, L. Schmidt, L. Alvisi, and M. Dahlin, “Exalt: Empowering Researchers to Evaluate Large-scale Storage Systems,” Proc. the 11th USENIX Conference on Network Systems Design and Implementation (NSDI 14), pp. 129-141, 2014.

[14] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Scars, “Benchmarking Cloud Serving Systems with YCSB,” Proc. the 1st ACM Symposium on Cloud Computing, pp. 143-154, 2010.

[15] M. Wittle, and BE. Keith, “LADDIS: The Next Generation in NFS File Server Benchmarking,” USENIX Summer, pp. 111-128. 1993.

[16] I. DRAGO, E. Bocchi, M. Mellia, H. Slatman, and A. Pras, “Benchmarking personal cloud storage,” Proc. the 2013 conference on Internet measurement conference. ACM, pp. 205-212. 2013.

[17] W. Sobel, S. Subramanyam, A. Sucharitakul, J. Nguyen, H. Wong, A. Klepchukov, S. Patil, A. Fox, and D. Patterson, “Cloudstone:

Multi-platform, multi-language benchmark and measurement tools for web 2.0," Proc. CCA, Vol. 8, 2008.
[18] J. Lee, C. Song, and K. Kang, "Benchmarking Large-scale Object Storage Servers," Proc. 40th IEEE Annual Conference on Computers, Software and Applications (COMPSAC), pp. 594-595, 2016.
[19] YCSB - v0.10.0 Release note. [Online]. Available: https://github.com/brianfrankcooper/YCSB/releases/tag/0.10.0
[20] NRDC, "America's Data Centers Are Wasting Huge Amounts of Energy," 2014.
[21] 정상진, 김문구, "그린 데이터센터 표준화 및 인증제 동향," TTA Journal, Vol. 145, pp. 76-82, 2013.

저 자 약 력

박 찬 영

이메일 : chanyoung@hanyang.ac.kr

- 2016년 한양대학교 컴퓨터공학과 (학사)
- 2016년~현재 한양대학교 컴퓨터공학과 석박사통합과정
- 관심분야 : 저전력 스토리지 시스템, 차량 시스템, 운영체제

이 재 면

이메일 : jaemyoun@hanyang.ac.kr

- 2012년 한양대학교 컴퓨터공학과 (학사)
- 2012년~현재 한양대학교 컴퓨터공학과 석박사통합과정
- 관심분야 : 지능형 모바일 컴퓨팅 플랫폼, 저전력 스토리지 시스템, 사이버피지컬 시스템

강 경 태

이메일 : ktkang@hanyang.ac.kr

- 1999년 서울대학교 수학계산통계학과군 전산과학전공 (학사)
- 2001년 서울대학교 전기컴퓨터공학부 (석사)
- 2007년 서울대학교 전기컴퓨터공학부 (박사)
- 2008년 PostDoc. Res. Ass. Coordinated Science Lab., UIUC
- 2010년 PostDoc. Res. Ass. Dept. Computer Science, UIUC
- 2011년~현재 한양대학교 컴퓨터공학과 부교수
- 관심분야 : 사이버피지컬 시스템, 모바일 컴퓨팅, 운영체제

공공 빅데이터 플랫폼 이용 활성화 전략

김지용 (정부통합전산센터), 김대엽 (펜타시스템 테크놀로지)

목 차

1. 서 론

정부의 운영패러다임이 '정부 1.0(정부 일방향) → 정부 2.0(국민 · 정부 양방향) → 정부3.0(국민 · 정부 개인맞춤형)'으로 전환되었다. 정부3.0은 정부가 신뢰받는 정부, 국민행복이란 비전을 실현하기 위해 국민 개개인에게 맞춤형 서비스를 제공하는 노력에 최대 역점을 두는 새로운 정부의 운영 패러다임을 말한다. 아울러 '데이터에 기반한 과학적 정책수립' 지원을 위해 정부3.0 추진위원회, 정부부처 및 지자체 등에서 빅데이터 관련 업무를 수행하고 있다. 공공분야에서는 행정자치부가 빅데이터 관련 법 · 제도 정비 및 빅데이터 사업을 발굴하고 있으며 행정자치부 소속기관인 정부통합전산센터에서는 범정부 기반의 빅데이터 플랫폼을 구축하여 정부 또는 지자체에서 수행하는 빅데이터 분석 업무를 지원하고 있다. 민간분야에서는 미래창조과학부가 민간 빅데이터 활성화를 위해 빅데이터센터를 운영하고 있으며, 이를 통해 창업자, 중소벤처, 대학 등에 Shared Service를 제공함으로써 산업 활성화를 촉진하고 있다. 더불어 일부 지자체에서도 빅데이터 전문조직을 구성하고 있으며 빅데이터 활용 관련 조례를 제정하고 있다.

정부3.0 추진위원회와 행자부가 조사한 공공데이터 개방 국민인식 조사결과('15년 9월~10월) 공공데이터 개방정책에 대해서는 약 64%가 인지하는 것으로 나타났으나, 개방되는 공공데이터의 양이 전반적으로 부족하다는 답변이 약 51%였다. 한편 공공데이터 개방 시 고려되어야 할 사항으로 '가치 있는 공공데이터 제공(37%)'과 공공데이터의 정확성, 최신성 및 표준화 등을 뽑았으며, 공공데이터 이용을 위한 정부의 지원책으로 50%이상이 교육제공과 IT 인프라 제공을 뽑았다.

이에 공공분야에서는 정부통합전산센터가 빅데

※ 본 내용은 필자의 주관적인 의견이며, 정부통합전산센터의 공식적인 입장이 아님을 밝힙니다.

이터 관련 IT 인프라 및 플랫폼 교육을 제공하고 있으며, 민간분야에서는 미래창조과학부가 그 역할을 하고 있다. 본 논문에서는 정부통합전산센터에서 운영 중인 빅데이터 공통기반 플랫폼을 중심으로 공공 빅데이터 플랫폼 이용활성화 전략을 논하고자 하며 논문의 구성은 다음과 같다. 2절에서는 공공 빅데이터 플랫폼의 전반적인 운영현황을 소개하고, 3절에서는 공공 빅데이터의 분석현황 및 한계점을 논한다. 마지막 절에서 공공 빅데이터 플랫폼 이용 활성화 전략을 제언한다.

2. 공공 빅데이터 플랫폼 운영현황

정부통합전산센터(이하 '센터')에서는 부처·지자체 공무원이 공동으로 활용할 수 있는 '범정부 공통기반 플랫폼'(이하 '혜안', www.insight.go.kr)을 행정 업무망을 통해 운영하고 있으며, 플랫폼 구성은 (그림 1)과 같다. 혜안에서는 외부 공공기관정보와 민간정보를 Open API 및 크롤링(Crawling) 등의 방법으로 수집·연계하여 센터 내에 저장하고, 빅데이터 처리·분석·시각화 도구를 활용하여 분석한 결과물을 관리 및 제공하며, 제공된 결과물은 중앙부처·지자체의 정책수립에 활용될 수

(그림 1) 빅데이터 공통기반 플랫폼 구성도

있다. 혜안의 주요 서비스로는 누구나 즉시 서비스를 이용할 수 있는 언론스크랩과 실시간으로 변화하는 화제의 키워드 관련 연관 검색어의 추이 분석 결과를 제공하는 실시간 키워드 분석서비스 등 일반분석 서비스가 있으며, 전문 분석가가 대용량 데이터를 처리·분석·시각화 할 수 있도록 필요한 도구를 제공하는 전문분석 서비스가 있다.

아울러 센터에서는 빅데이터 분석능력이 부족한 공무원도 혜안을 활용하여 직접 분석할 수 있도록 '빅데이터 공통기반 활용모델'을 제작하여 빅데이터 분석 이용활성화에 앞장서고 있다. 주요 모델로는 위치기반 분석 모델, 소셜 분석 모델, 홈페이지 게시글 분석 모델이 있다. 위치기반 분석 모델은 국민생활과 밀접하게 연관된 지역·장소 데이터를 기존의 분석방식(통계표, 차트 등)에서 벗어나 GIS를 통한 위치 기반의 데이터분석 결과로 도출하는데 사용된다. 소셜 분석 모델은 센터가 수집하고 있는 소셜 데이터와 결합하여 여론동향을 파악하거나 정책방향을 설정하는데 적극 활용되고 있으며 홈페이지 게시글 분석 모델은 행정기관이 보유한 홈페이지 게시글의 비정형 데이터를 워드 클라우드, 랭킹, 소셜네트워크(SNA) 및 열지도(Heatmap) 등으로 결과를 표출하여 그 결과를 민원분석에 활용하고 있다.

3. 공공 빅데이터 분석현황 및 한계

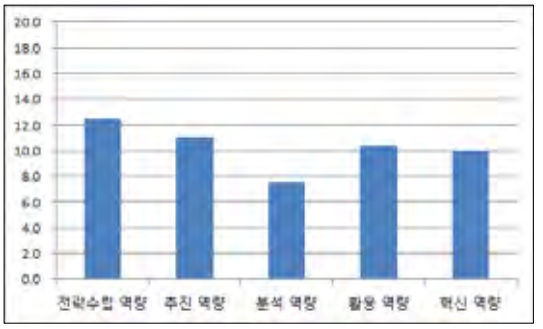
빅데이터 분석을 성공적으로 수행하기 위해서는 우선 분석의 목적과 목표를 명확하게 정의해야 하며, 보유하고 있는 데이터에 대해 충분히 이해하고 있어야 한다. 또한 데이터를 어떤 형태로 가공하고, 가공된 데이터를 어떠한 기법을 활용하여 분석할 것인지에 대한 전략도 미리 수립되어야 하며 분석 결과를 어떠한 형태로 시각화하고 활용할 것인지에 대한 계획도 구상해야 한

다. 즉, 빅데이터를 분석하고 활용하기 위해서는 데이터에 대한 이해와 더불어 데이터 가공 및 분석 기법, 분석결과 활용 전략 등과 관련된 전반적인 지식과 노하우가 요구된다.

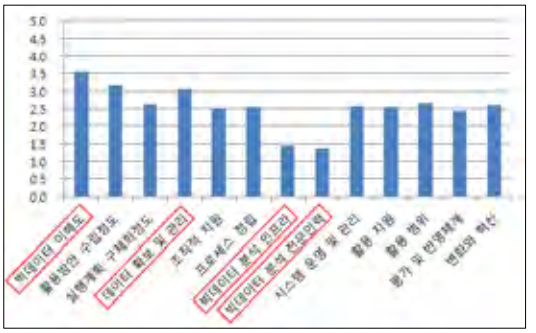
하지만 대부분의 정부부처나 지자체 공무원은 빅데이터 분석 방법 및 활용에 대한 지식과 노하우가 매우 부족한 것이 현실이다. 센터에서는 정부부처·지자체 공무원의 빅데이터 분석 역량을 향상시키기 위해 올해 두 차례의 빅데이터 분석 전문교육을 실시하였으며, 각 기관의 빅데이터 분석 역량을 파악하기 위한 설문조사를 실시한 바 있다. 설문조사 내용은 미래창조과학부가 주관하고 한국정보화진흥원 등이 공동으로 개발한 빅데이터 분석역량 진단모델을 기반으로 구성하였다. 설문 결과에 따라 기관의 역량 수준은 초기(initial), 인식기(repeatable), 정립기(defined), 관리기(managed), 혁신기(innovated)의 5단계 중 하나로 결정된다. 분석 역량 수준의 결정은 설문을 구성하는 다섯 가지의 역량(전략수립 역량, 추진 역량, 분석 역량, 활용 역량, 혁신 역량) 점수에 따라 결정되며, 이들 각각의 역량은 다시 그것을 구성하는 중항목 점수에 따라 결정된다.

설문조사에 참여한 중앙부처·지자체 공무원의 응답 결과를 종합해 본 결과 약 90% 정도의 기관이 인식기(repeatable) 이하의 역량 수준을 보이고 있으며, 상위에 속한 기관 역시 핵심 역량(빅데이터 분석 인프라, 빅데이터 분석 전문인력 등) 부분에서는 다소 미흡한 수준을 보이고 있다. (그림 2)와 (그림 3)은 설문에 참여한 기관 공무원에 대해서 다섯 가지 역량과 그것을 구성하는 세부 항목들의 평균 측정치를 보여주고 있다.

대부분의 기관이 빅데이터에 대한 이해도나 데이터의 확보 및 관리 차원에서는 어느 정도 준수한 수준의 역량을 보유하고 있지만, 빅데이터 분석 인프라 구축 및 빅데이터 분석 전문 인력



(그림 2) 빅데이터 핵심 역량에 대한 평균 점수



(그림 3) 세부 역량에 대한 평균 점수

확보 등의 핵심 역량에서는 기대에 못 미치는 수준을 보이고 있다. 다시 말하면 빅데이터 분석의 필요성을 인식하고는 있지만 실질적인 분석 역량의 확보에서는 아직 미진한 부분이 많다는 것을 알 수 있다.

정부부처·지자체 공무원의 빅데이터 분석 역량이 데이터 기반의 과학적 행정을 구현하는데 있어 아직 많은 한계를 보이고 있는 가운데 센터에서는 빅데이터 공통기반 시스템 구축, 분석 서비스 제공, 분석과제 지원 및 전문교육 실시 등의 역할을 수행하고 있지만, 빅데이터 활성화 측면에서 여전히 많은 문제와 제약사항이 존재한다. 현존하는 문제점을 정리해보면 다음과 같다.

첫째, 대부분의 부처 및 지자체는 각 기관에서 보유한 데이터에 대해서 자체적으로 분석을 수행하기보다 센터와 같은 빅데이터 전문기관에

의존하는 경우가 대부분이며, 경우에 따라 별도의 분석사업을 발주하여 외부 업체에 위탁하기도 한다. 이러한 방법은 시간이나 비용 측면에서 효율성을 저하시키는 요인이 될 수 있다. 빅데이터 활용의 활성화를 위한 근본적인 방법은 각 기관이 빅데이터 전문 인력을 적극 양성하고 필요시 언제든지 빅데이터 분석을 수행하고 정책 마련에 활용할 수 있도록 하는 것이다.

둘째, 부처나 지자체에서 센터에 빅데이터 분석을 의뢰하는 경우를 보면, 해당 기관장의 관심사항으로서 일회성 분석으로 진행되는 것을 종종 볼 수 있다. 이와 같은 경우는 대부분 분석 요구사항이 명확하지 않거나 막연하게 분석을 요청하는 경향이 빈번하게 나타난다. 앞서 언급한 바와 같이 빅데이터 분석의 활용 효과를 높이기 위해서는 분석의 목적과 목표가 명확하게 정의되어야 한다. 만약 그렇지 않으면 분석의 활용도는 저하되고 단순히 이벤트성 과제로 끝날 가능성이 높아질 것이다.

셋째, 낯선 빅데이터 분석환경 또는 기법(R 프로그래밍, SNA분석 등)으로 인해 직접 빅데이터 분석을 시도하는데 많은 부담을 느끼는 경향이 있다. 빅데이터 분석은 통계 분야에 대한 지식과 고수준의 데이터 처리 기술 등을 요구한다. 해당 분야에 경험이 부족한 공무원은 빅데이터 분석에 어려움을 겪을 수밖에 없다. 이러한 한계를 해결하기 위해서는 빅데이터 분석에 대한 꾸준한 교육과 다양한 사례 공유 등이 필수적으로 요구된다.

마지막으로 빅데이터 분석의 경우, 다양한 출처의 데이터를 융·복합함으로써 그 효과를 극대화시킬 수 있다. 즉, 데이터 기반의 과학적 행정을 실현하기 위해서는 다양한 기관의 데이터들을 서로 연계하고, 그로부터 분석을 통한 새로운 인사이트(insight)를 확보해야 한다. 하지만 각 기관의 데이터를 확보 및 연계할 수 있는 법적 근거가 부족한 관계로 분석에 필요한 데이터가 부족한 것이 가장 큰 걸림돌로 작용한다. 업무협의를 통해 어렵게 확보된 데이터 역시 정확성이 떨어지고 표준화되지 않은 형태로 존재하기 때문에 효과적인 분석에 어려움으로 작용한다.

지금까지 살펴본 바와 같이, 국내 공공기관의 빅데이터 분석 환경은 매우 열악하며, 이를 개선하기 위한 보다 근본적인 시도와 노력이 필요하다. 비단 한 기관만의 노력이 아닌 범정부적인 문제인식과 해결 노력이 수반되어야 한다.

4. 공공 빅데이터 플랫폼 이용 활성화 전략

올해 각 중앙부처·지자체에서는 기관의 고유업무를 기반으로 다양한 분야에서 빅데이터 관련 추진계획이 마련되고 있다. 이에 따라 본 고에서는 기 구축된 혜안시스템을 이용해 데이터 기반의 과학적·선제적 정책수립 지원 및 분석 전문인력 양성을 위해 센터가 선도적 역할을 할 수 있는 부문에 한정해 아래와 같이 활성화 제고 방안을 마련코자 한다.

첫째, 행정기관 등의 장은 전자정부법에 따라 다른 행정기관 등과의 상호연계 또는 공동이용과 관련한 전자정부사업 및 지역정보화사업을 추진할 때 중복투자 방지 등을 위하여 중앙사무관장기관의 장과 사전에 협의하여야 한다고 명시되어 있으나, 올해 상반기에 추진되는 정부부처·지자체별 빅데이터 사업 대부분이 플랫폼 구축사업이거나 여론 동향분석을 위한 서비스로 이는 센터에서 운영중인 플랫폼 및 빅데이터 서비스와 상당히 유사하다. 따라서 센터에서 기 구축한 플랫폼을 중앙부처·지자체에서 활용하도록 유도하기 위해 사전협의를 통한 빅데이터 사

업의 중복성 검토 과정을 강화하거나 중앙부처·지자체에서 추진하는 빅데이터 관련 사업을 센터와 먼저 협의할 수 있는 절차 등을 마련해야 한다.

둘째, 정부에서는 정부업무평가기본법에 따라 매년 행정관리역량평가 및 지자체 합동평가를 실시하며, 평가지표를 통해 국정의 효율적인 업무수행을 평가한다. 센터에서도 중앙부처·지자체에서 혜안시스템을 활용한 빅데이터 분석업무를 적극 추진할 수 있도록 평가지표를 마련하여 데이터 기반의 과학적 정책을 수립할 수 있도록 지원하여야 한다.

셋째, 빅데이터 분석의 이용활성화를 위해 센터에서는 기 구축한 3종의 공통기반 활용모델 이외에 중앙부처·지자체의 업무 수요를 파악해서 상시적으로 활용할 수 있는 공통기반 활용모델을 적극 발굴하여야 한다.

넷째, 최근 공공분야에서는 기상·기후 빅데이터 플랫폼(기상청)을 통해 기상·기후 데이터와 타분야 데이터를 융합·분석할 수 있는 환경을 제공하고 있으며, 빅데이터 캠퍼스(서울시) 등을 통해 창업 및 정책 등의 아이디어 발굴을 지원하고 있다. 민간분야 역시 누구나 빅데이터 플랫폼을 활용하여 국민의 라이프스타일 트렌드를 분석하고 그 결과를 다양한 측면에서 활용할 수 있는 플랫폼 개방이 진행되고 있다. 따라서 혜안시스템의 이용자 확대 및 다양한 분석지원을 위해 행정 업무망에서만 이용되는 시스템을 정부·지자체 산하의 연구소 직원이 활용할 수 있도록 개방하는 추진계획을 마련해야 한다.

다섯째, 센터에서는 빅데이터 분석역량 강화를 위해 매년 상하반기 전문교육을 실시하고 있으나 전문 분석도구 및 공통기반 활용모델의 실습시간이 제한된 관계로 단시간 내에 혜안시스템을 이해시키기 어렵다. 따라서 온라인으로 언제, 어디서나 상시적으로 서비스를 제공받을 수 있도록 지원하여야 하며, 이를 공무원 교육 전문기관과 연계해서 온라인 혜안시스템 강좌를 적극 홍보해야 한다.

마지막으로 정부부처·지자체에서 추진하려는 빅데이터 분석사업 및 플랫폼 구축사업은 일반적인 정보시스템 구축사업과 상이하며 공공기관의 분석사례 또한 부족한 상황이다. 따라서 센터에서는 분석전문 인력을 활용하여 수요기관에 맞춤형 컨설팅을 제공할 수 있어야 하며, 기 구축된 분석결과(서비스)를 혜안시스템을 통해 공유하여 다양한 아이디어가 도출될 수 있도록 적극 지원하여야 한다.

5. 결 론

지금까지 혜안시스템의 운영현황과 중앙부처·지자체 공무원이 빅데이터 분석 시 당면하는 애로사항 및 이에 대한 이용 활성화 제고방안을 알아보았다. 올 상반기 정부에서는 빅데이터 관련 30여종의 보도자료를 배포하였고 주요 내용은 빅데이터를 활용한 대민서비스 지원이다. 공공 빅데이터 서비스를 추진하기 앞서, 정부부처·지자체에서는 다음 사항을 유의하여야 한다.

첫째, 데이터에 기반한 공공서비스를 일정 수준이상으로 국민에게 제공하기 위해 데이터의 품질을 정의하고 객관적인 측정기준을 마련해야 한다. 특히 국민 중심의 맞춤형 서비스 구현을 위해서는 서비스 품질에 대해 상시 측정하고 그 품질을 향상시키려는 노력이 절실하다.

둘째, 중앙부처·지자체에서는 데이터 사이언티스트 인력확보가 중요하다. 이는 빅데이터를 도입하는 기관이 추진전략 및 수행계획을 수립하는데 있어 유의미한 분석 결과가 필요하기 때문에 데이터 사이언티스트의 역할이 매우 중요

하다고 볼 수 있다.

구글의 알파고(AlphaGo)와 이세돌의 바둑 대국이후, ‘AI(Artificial Intelligence)’ 에 대한 관심이 급증하고 있다. 정부에서도 빅데이터와 인공지능 기술의 결합을 통한 미래예측 시스템 구축 및 신규 지식 서비스 제공 방안을 마련하고 있다. 인공지능에서 우수한 판단을 하기 위해서는 데이터가 필수이다. 빅데이터의 궁극적인 목표는 인공지능의 완성일 수 있다. 빅데이터 기술이 주목받는 이유는 정보처리에 있지 않으며 우수한 정보처리를 바탕으로 의미있는 결과를 도출할 수 있는 점이 빅데이터가 주목받는 이유이다. 이에 따라 센터에서는 전문교육 과정을 통해 분석전문가를 양성하고 분석역량 강화를 위해 가이드 북을 제작 및 배포하였다. 또한 다양한 분석사례를 통해 기관에서 필요로 하는 분석 아이디어를 도출할 수 있도록 지원체계를 마련하였다. 향후 더욱 강화된 지원체계를 통해 중앙부처・지자체 공무원이 혜안시스템을 이용할 수 있도록 지원할 예정이다.

참 고 문 헌

[1] 행정자치부, www.moi.go.kr

[2] 정부3.0, www.gov30.go.kr

[3] 빅데이터 공통기반 플랫폼(혜안), www.insight.go.kr

[4] 국가법령정보센터, www.law.go.kr

[5] 정부3.0 추진위원회, 정부3.0-데이터 기반의 과학적 행정, 2016년 4월

[6] 한국정보기술학회지, 빅데이터가 인공지능에 미친 영향, 2016년 제14권1호

[7] 행정자치부, 공공서비스 수준진단 및 품질제고 방안 연구용역 결과보고서, 2016년 8월

[8] 정보통신산업진흥원, 빅데이터 추진전략을 위한 제언, 2014년 5월

[9] 빅데이터 분석활용센터, 빅데이터 역량 진단모델, 2014년 2월

저 자 약 력

김 지 용

이메일 : coolguy@korea.kr

- 2008년 고려대학교 소프트웨어공학과 (석사)
- 2004년~서울시, 국립과학수사연구원, 행정자치부, 정부통합전산센터
- 관심분야 : 정보시스템 표준화, EA

김 대 엽

이메일 : dykim@penta.co.kr

- 2015년 충남대학교 컴퓨터공학과 (박사)
- 2005년 충남대학교 컴퓨터공학과 (학사)
- 2016년~펜타시스템테크놀러지 책임컨설턴트
- 관심분야 : 객체지향 분석 및 설계, 컴포넌트기반 개발방법론, 소프트웨어 품질관리, 영향분석, 기능점수, 빅데이터

정보처리학회지 2016년도 5월호 게재 목차

■ 2016년 5월 (제23권 제3호) ■ 특집명 : 개방형 IoT

정보처리학회논문지 2016년도 5월호 게재 목차

■ 제5-CCS권 제5호(통권 제44호) 2015년 5월

■ 제5-SDE권 제5호(통권 제44호) 2016년 5월

정보처리학회논문지 2016년도 6월호 게재 목차

■ 제5-CCS권 제6호(통권 제45호) 2016년 6월

■ 제5-SDE권 제6호(통권 제45호) 2016년 6월

JIPS(정보처리학회영문지) 2016년도 6월호 게재 목차

■ Volume 12, Number 2(Serial Number 40), June, 2016

학회동정

[학회 주최/주관 행사]

◈ 2016년도 제 2차 단기강좌 개최

1. 일 시 : 2016년 5월 20일(금) 09:30
2. 장 소 : CNN the Biz 교육연수센터 401호
3. 참석자 : 45명(일반 21명, 학생 24명)
4. 내 용 : 사물인터넷 최신 기술 동향 및 핵심 구현 기술

[2016년도 제2차 단기강좌에서 한연희 프로그램위원장의 인사말 모습]

[2016년도 제2차 단기강좌 개최 모습]

◈ 2016년도 IT 21 글로벌 컨퍼런스 개최

1. 일 시 : 2016년 6월 9일(목) ~ 10일(금)
2. 장 소 : 섬유센터 17층 컨퍼런스홀
3. 주 제 : 소프트웨어 중심 초연결 사회 구현
4. 위원회 :
   - 공동조직위원장 – 박청원 원장(KETI), 백기승 원장(KISA), 서병조 원장(NIA), 윤종록 원장(NIPA), 이상훈 원장(ETRI), 한선화 원장(KISTI)
   - 프로그램위원장 – 김상훈 교수(한경대)
   - 산학협력위원장 – 이태하 대표이사(대우정보시스템)
   - 공동홍보위원장 – 윤찬현 교수(KAIST), 한근희 교수(고려대), 김정억 부장(전자신문), 원유재 교수(충남대)
   - 공동운영위원장 – 이주연 교수(아주대), 문남미 교수(호서대), 윤명현 본부장(KETI), 이필우 본부장(KISTI)
5. 내 용 : 초청강연 4개, 트랙별 발표(7개 36개 세션), 리셉션
6. 참석자 : 302명(일반 199명, 학생 103명)
7. 초청강연 : 김용채 대표이사(이온리얼리티코리아)
   이성환 교수(고려대)
   박종목 이사(네이버랩스)
   장병탁 교수(서울대)
8. 트랙주제 : 사물인터넷 소프트웨어 및 플랫폼
   빅데이터와 인공지능
   AR / VR 요소 기술 및 응용
   주요연구기관 세션
   오픈소스와 클라우드 컴퓨팅
   스마트 팩토리
   자율주행과 스마트카

[2016년도 IT 21 글로벌 컨퍼런스 초청강연 1 모습
– 김용채 대표이사(이온리얼리티코리아)]

[2016년도 IT 21 글로벌 컨퍼런스 개회식에서
장석영 국장(미래창조과학부)의 축사 모습]

[2016년도 IT 21 글로벌 컨퍼런스 초청강연 2 모습
– 이성환 교수(고려대학교)]

[2016년도 IT 21 글로벌 컨퍼런스 개회식에서
한선화 원장(KISTI)의 환영사 모습]

[2016년도 IT 21 글로벌 컨퍼런스 개회식에서 구원모 회장의
개회사 모습]

[2016년도 IT 21 글로벌 컨퍼런스 개회식에 참석한 내빈 모습]

[2016년도 IT 21 글로벌 컨퍼런스 1일차 트랙별 발표 모습]

[2016년도 IT 21 글로벌 컨퍼런스 리셉션에서 공로상 수여 모습 - 이태하 대표(대우정보시스템)]

[2016년도 IT 21 글로벌 컨퍼런스 리셉션에서
구원모 회장의 인사말 모습]

[2016년도 IT 21 글로벌 컨퍼런스 리셉션에서
박두순 전임회장의 건배사 모습]

[2016년도 IT 21 글로벌 컨퍼런스 리셉션에서
정진욱 전임회장의 축사 모습]

[2016년도 IT 21 글로벌 컨퍼런스 리셉션에 참석한 임원 모습]

[2016년도 IT 21 글로벌 컨퍼런스 2일차 트랙별 발표 모습]

[공동 주최/주관 행사]

◈ 제235회 스마트 사회 지도자 포럼 개최

1. 일 시 : 2016년 5월 13일(금) 07:00
2. 장 소 : 밀레니엄힐튼호텔 B1 그랜드볼룸
3. 주 관 : 도산아카데미 공동 주관
4. 참석자 : 곽덕훈 회장 외 40여명
5. 강연자 : 이성환 교수(고려대학교 뇌공학과)
6. 제 목 : 알파고와 인공지능

[제235회 스마트 사회 지도자 포럼 개최 모습]

◈ 제236회 스마트 사회 지도자 포럼 개최

1. 일 시 : 2016년 6월 3일(금) 07:00
2. 장 소 : 밀레니엄힐튼호텔 B1 그랜드볼룸
3. 주 관 : 도산아카데미 공동 주관
4. 참석자 : 곽덕훈 회장 외 40여명
5. 강연자 : 박수용 원장(글로벌핀테크연구원)
6. 제 목 : 4차 산업 혁명과 핀테크

[제 236회 스마트 사회 지도자 포럼 개최 모습]

[제회의]

◈ 2016년도 제 2차 회장단회의 개최

1. 일 시 : 2016년 5월 20일(금) 오후 6시(18:00)
2. 장 소 : CNN the Biz 교육연수센터 401호
3. 참석자 : 정영식 수석부회장 외 16명
4. 내 용 : 2016년도 IT 21 글로벌 컨퍼런스 진행사항 점검

[2016년도 제2차 회장단회의에 참석한 임원 모습]

[2016년도 제2차 회장단회의 개최 모습]

[각위원회 회의]

– 이브릿지연구회

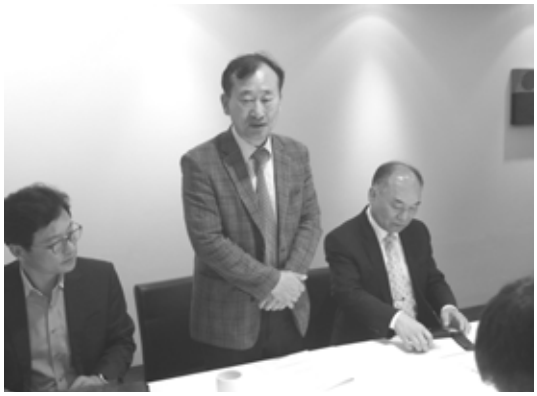
◈ 2016년도 제3차 이브릿지편집위원회 회의 개최

1. 일 시 : 2016년 6월 14일(화) 11:00
2. 장 소 : 달개비
3. 참석자 : 안문석 위원장 등 14명
4. 내 용 : 강은희 장관(여성가족부) 축하패 전달 및 제 21호 발간 협의

[2016년도 제3차 이브릿지편집위원회에서
강성주 정책관(미래창조과학부)의 인사말 모습]

[2016년도 제3차 이브릿지편집위원회에서
안문석 편집위원장의 인사말 모습]

[2016년도 제 3차 이브릿지편집위원회에서
강은희 장관(여성가족부)의 축사 모습]



[2016년도 제3차 이브릿지편집위원회에서
이정배 이브릿지연구회위원장의 인사말 모습]

[2016년도 제3차 이브릿지편집위원회에서 감사패 수여 모습
– 강은희 장관(여성가족부)]

[2016년도 제3차 이브릿지편집위원회 회의 개최 모습]

[2016년도 제3차 이브릿지편집위원회에 참석한 내빈 모습]

신규회원 명단

2016년 6월 1일~7월 31일

회원구분	회원번호	성명	직장명
종신회원	2016-20830-01	김석훈	순천향대학교
	2016-20833-01	민석홍	시스템뱅크
	2016-20829-01	박진호	숭실대학교
갱신된 종신회원	2014-19515-01	신상규	산업기술대학원대학
	2010-16541-01	임경수	한국전자통신연구원
	2012-18382-01	임종범	고려대학교
정회원	2016-20816-02	조형주	경북대학교
	2016-20820-02	김정준	한국산업기술대학교
	2016-20817-02	최우용	동아대학교
	2016-20818-02	정용현	고려대학교
	2016-20824-02	강주영	고려대학교

회원구분	회원번호	성명	직장명
준회원	2016-20819-03	김준용	고려대학교
	2016-20827-03	김재기	고려대학교
	2016-20821-03	윤의녕	인하대학교
	2016-20823-03	박한곤	아주대학교
	2016-20825-03	백현우	고려대학교
	2016-20826-03	우만균	한국정보통신
	2016-20828-03	최지은	숙명여자대학교
	2016-20831-03	민상훈	숭실대학교
	2016-20832-03	방승규	고려대학교

특별 법인회원 명단

구 분	대표자	주 소
(주)경봉	윤석원 대표	경기도 안양시 만안구 예술공원로 153-32
(주)베스트케이에스	김교은 대표	서울시 금천구 범안로 1130 가산디지털엠파이어빌딩 501, 502호
(주)블루코어	이동화 대표	서울시 강남구 역삼동 682 남전빌딩 4층
삼성SDS(주)	정유성 대표	서울시 송파구 올림픽로35길 123(신천동) 삼성SDS타워
(주)영화조세통람	서동혁 대표	서울시 중구 동호로 14길 5-6 이나우스빌딩
(주)LG CNS	김영섭 대표	서울시 영등포구 여의대로 24, FKI타워
(주)자이네스	고범석 대표	서울시 구로구 디지털로33길 55 904호(E&C벤처드림타원 2차)
정보통신산업진흥원	윤종록 원장	충북 진천군 덕산면 정통로 10
정보통신정책연구원	김도환 원장	충북 진천군 덕산면 정통로 18
(주)지란지교시큐리티	윤두식 대표	서울시 강남구 역삼로 542(대치동 신사S&G 5층)
(주)G.I.G기업	이용기 대표	서울시 광진구 능동로40길8 정암빌딩 100호
KCC정보통신	이상현 대표	서울시 강서구 공항대로 665 KCC오토타워(염창동 260-4번지)
한국인터넷진흥원	백기승 원장	서울시 송파구 중대로 135 IT벤처타워 4층
한국정보화진흥원	서병조 원장	대구시 동구 첨단로 53
한국전자통신연구원	이상훈 원장	대전시 유성구 가정로 218

한국정보처리학회 기관지 원고 집필 안내

한국정보처리학회는 학회지 「정보처리학회지」와 논문지 「정보처리학회논문지A·B·C·D」를 발행하고 있습니다. 「정보처리학회지」는 새로운 기술동향을 비롯해서 각종 정보를 게재하고, 회원의 지식 향상을 목적으로 하며, 「정보처리학회논문지A·B·C·D」는 회원의 연구 결과를 발표하는 장입니다.

본 안내는 학회 기관지의 원고 집필 요령을 정리한 것으로, 집필 시 참고로 하시기 바랍니다.

『정보처리학회지』 원고 집필 안내

제 1 조 학회지에 게재할 원고의 종류는 특집, 특별기고, 기획기사, 정보 관련 기술 동향 및 편집위원회가 인정하는 것으로 한다.

제 2 조 투고자는 원칙적으로 본 학회 회원으로 한다. 단, 회원과의 공동기고자 및 초청기고자는 예외로 한다.

제 3 조 원고는 수시로 접수하며 접수일은 원고가 본학회 편집위원회에 도착한 날로 하고, 접수된 원고는 편집위원회에서 게재여부를 결정한다.

제 4 조 원고는 가장 많이 사용되는 워드프로세서로 작성한 파일을 함께 제출한다.

제 5 조 원고의 내용은 정보처리 관련자가 이해할 수 있는 정도로 작성한다.

제 6 조 투고자는 200자 이내의 약력을 제출하여야 한다. 게재가 확정된 원고에 대해서는 추후 저자의 사진을 제출해야 한다.

제 7 조 본 학회지에 게재된 내용은 본 학회의 승인없이 영리목적으로 무단 복제하여 사용할 수 없다.

제 8 조 원고 작성 방법은 다음과 같다.

(1) 1페이지 기술 분량 : A4용지 30행×40자 내외

(2) 원고분량 : 6~8페이지 내외

(3) 참고문헌 : 참고 문헌은 저자명에 의한 사전식으로 기술하되, 각 참고 문헌은 잡지의 경우 “번호저자명, 제목, 잡지명, 권, 호, 페이지, 연도”의 순으로 기술한다. 단, 참고문헌 인용시에는 대괄호를 이용할 것(예 [1], [2], [3], [4] 등)

(예) [1] 김철수, 김수철, “한국 정보 처리 산업에 관한 연구”, 한국정보처리논문지, 제 1권, 제 1호, pp.23-43, 1997.

[2] 이영희, 컴퓨터입문, pp.234, 출판사, 1997.

[3] L. Lanomt, “Synchronization Architecture and Protocols”, IEEE Trans. on Comm., Vol. 23, No. 3, pp.123-132, 1997.

[4] Steinmetz, Multimedia : Computing, Communications & Applications, PII, 1995.

(4) 내용표기에 있어서, 장, 절 등의 표시는 ‘ 1, 1.1, 1.1.1, 가, 1), 가), (1), (가)’의 순서로 한다.

(5) 원고는 ‘제목-소속-성명-목차-본문-참고문헌’의 순으로 기술하며, 첫장 하단에는 회원 구분을 명기한다.

(6) 표의 제목은 “<표1>대한민국” 과 같이 표의 상단에 기술하고, 그림의 제목은 “(그림1)서울”과 같이 그림의 하단에 기술하며, 사진판으로 사용할 수 있도로 백지에 정서해야 한다.

본 규정은 1997년 1월 1일부터 효력을 발생한다.

기타 원고 모집 안내

당 학회지 편집위원회에서는 학회지 「정보처리학회지」에 게재할 각종 원고를 회원 여러분으로부터 모집하고 있습니다. 많은 투고와 참여있으시기 바랍니다.

1. 모집내용

다음에 대한 원고를 모집합니다.

(1) 해 설 : 정보처리에 관련된 신기술 또는 이론으로서 당 학회 회원의 관심도가 높은 내용
(2) 외국기사 : 외국 잡지에 게재된 기사로서 당 학회 회원에게 유익한 내용
(3) 서 평 : 최근에 출판된 책으로서 당 학회 회원에게 유익한 도서의 소개 또는 비평
(4) 뉴 스 : 정보처리에 관한 국제규모의 회의, 대회의 보고 등 시사성이 높고 당 학회 회원에게 널리 알릴 가치가 있는 내용
(5) 기관소개 : 국내 기관 또는 외국 기관
(6) 기 타 : 당 학회 회원에게 유익한 내용

2. 응모 자격

당 학회 회원으로 한다.

3. 응모 절차

원고는 학회지 편집위원회에서 정한 투고 규정에 의거하여 다음 순서로 기술하여 주시기 바랍니다.

(1) 제 목
서평의 경우에는 저자명, 책이름, 페이지수, 출판사, 발행년도, 가격 등으로 기술한다.
어느 장르에 속하는지를 첫페이지 오른쪽 상단에 표시한다.
(2) 필자명, 소속, 필자 연락처
(3) 본 문
본문은 서평의 경우 2,000자 정도, 뉴스의 경우 1,000자 정도로 한다.
(4) 참고문헌, 부록, 그림, 표
(5) 필자 소개
이름, 경력과 학력을 기술한다.

4. 원고 취급

투고된 원고는 학회지 편집위원회에서 심사를 한 후 게재여부를 결정합니다. 게재가 결정되었을 경우에는 원고 수정을 부탁하는 경우가 있습니다. 서평의 경우에는 필자의 사진이 필요하므로 게재 결정 후 학회 사무국으로 우송해야 됩니다.

5. 원고료

학회지 규정에 의거하여 소정의 원고료를 지급합니다.

6. 보낼 곳

140-750 서울특별시 용산구 한강대로 109, 1002호(한강로 2가 용성비즈텔)
한국정보처리학회 학회지 편집위원회
uskim@kips.or.kr

정보처리학회 논문지 투고 규정

1. 원고의 전자 투고

모든 원고는 전자 형태(MS Word, 아래아 한글, 혹은 PDF 형태)로 학술지 웹사이트 (http://acoms1.kisti.re.kr/kips/index.jsp?publisher_cd=kips&cid=&cid_year=2006&cid_seq=A&lang=kor)를 통해 온라인으로 투고하여야 한다. 투고 규정은 해당 웹사이트에서도 볼 수 있으며, 본 학술지에 투고하는 모든 원고들은 이 규정을 준수하여야 한다. 그렇지 않을 경우 원고가 반송되게 되며 이로 인해 출판이 지연될 수도 있다. 원고 투고에 관한 문의는 이메일(kips@kips.or.kr)이나 전화(+82-2-2077-1414), 팩스(+82-2-2077-1472)를 통해 학회 사무국으로 한다. 저자 중에 1인은 학회 회원으로 가입되어야 함을 원칙으로 한다.

2. 연구 및 출판 윤리

본 학술지는 Guidelines on Good Publication (http://publicationethics.org/static/1999/1999pdf13.pdf)에 기술된 연구 및 출판 윤리 지침을 따른다.

2.1 이해갈등관계 명시

저자는 기업으로부터의 재정적 지원 또는 연계, 이익집단으로부터의 정치적 압력 등과 같은 이해 갈등 관계가 있으면, 이에 관한 정보를 밝혀야 한다. 특히, 연구에 관계된 모든 지원금의 출처를 명백히 진술해야 한다.

2.2 저자 요건

1) 연구의 기본개념설정과 설계, 자료수집, 또는 자료분석과 해석에 지대한 공헌을 하고, 2) 원고를 작성하거나 내용의 중요 부분을 변경 또는 개선하고, 3) 최종 원고의 내용에 동의한 세 가지 조건을 모두 충족한 사람만이 논문 저자로서 원고에 나열되어야 한다. 원고의 최초 투고 후, 어떠한 저자 변경 사항(저자 추가, 저자 삭제, 혹은 저자 순서 변경)도 편집인에게 편지로 알려주고 승인을 받아야 한다. 이 편지에는 해당 논문의 모든 저자들의 서명이 포함되어야 한다.

2.3 이중게재/이중투고 금지

투고 된 모든 원고는 다른 학술지에 이미 실렸거나 또는 심사 중이어서는 안 된다. 채택된 원고의 모든 부분은 편집위원회의 허가 없이 다른 과학학술지에 이중게재 하여서는 안 된다. 본지에 실린 논문의 이중게재 발각 시에는 저자 및 소속기관에 이를 알릴 것이며, 저자에게 제재가 가해 질 것이다.

3. 상호심사 절차

모든 원고는 편집위원이 위촉한 2인 또는 3인의 심사위원들이 평가하며, 연구의 질과 독창성, 그리고 과학적 중요성을 바탕으로 심사하여 채택 여부를 결정한다. 원고투고 후 심사결과를 이메일로 통보 받게 되며 심사자의 의견이 교신저자의 이메일로 전달된다. 교신저자는 수정된 원고를 온라인으로 재투고해야 하며 심사자의 지적에 따라 변경된 내용을 각 항목별로 진술해야 한다. 편집위원회 결정 이후 8주가 경과해도 수정된 원고를 재투고하지 않을 시에는 철회로 간주한다. 저자는 학술지 웹사이트에서 투고 논문의 심사 진행 현황을 확인할 수 있다.

4. 저작권

출판된 모든 원고는 한국정보처리학회의 자산이 되며, 서면허가 없이 다른 곳에 출판되어서는 안 된다. 출판이 결정되면 저자는 저작권양도 서식을 기재하여 팩스, 우편 또는 이메일로 학회 사무국에 보내야 한다.

5. 원고 작성

5.1 언어

모든 원고는 국문 또는 영문으로 작성하여야 한다. 국문 논문의 경우, 서지 정보(제목, 저자, 소속, 교신저자의 주소와 이메일), 표, 그림, 감사의 글, 참고문헌 등은 모두 영문으로 기술하여야 한다. 심사를 위한 초기 투고 원고에는 저자 정보를 포함시키지 말아야 한다. 하지만, 논문 수락 판정을 받은 후 제출하는 최종본에는 저자 정보를 포함시켜야 한다.

5.2 일반적인 사항

1) 원고는 MS Word나 한글문서로 작성한다.

2) 원고는 A4 (21.0×29.7cm) 용지에 10point 글씨크기로 행 사이를 2행 간격(double space)으로 하여 작성하되, 상하좌우 모두 2.5cm의 여백을 둔다.

3) 모든 단위는 International System(SI) of Units 에 따라 기술하여야 한다. 퍼센트(%)와 온도(℃)를 제외한 모든 단위는 한 칸의 공백 다음에 기술해주어야 한다.

5.3 출판 유형

한국정보처리학회논문지는 연구논문(research paper), 편집인의 글(editorial), 편집인과의 서신(letters to the editor) 등을 출판한다.

1) 연구논문(research paper): 본 학술지가 다루는 범위 안에서 새로운 학술적 발견들을 상호 심사과정을 거쳐 연구논문으로 출판할 수 있다. 연구논문에는 이론이나 실험에 관한 새롭고 중요한 결과들이 기술되어야 한다. 연구논문 중 일반논문(regular paper)과 단편논문(short paper)의 길이 제한은 각각 20쪽과 4쪽 이내이다.

2) 편집인의 글(editorial): 편집인의 글은 초빙에 의해서만 원고를 투고할 수 있으며, 본 학술지 편집위원회에서 결정하는 주제들을 다룬다.

3) 편집인과의 서신(letters to the editor): 본 학술지에 이미 출판된 학술 논문에 관한 간략한 평가나 흥미로운 새로운 아이디어를 편집인과의 서신으로 투고할 수 있다. 학술지 편집위원회에서는 투고된 서신을 편집할 수 있으며, 필요한 경우 해당 논문의 저자에게 회신을 요청할 수도 있다.

5.4 연구논문

원고는 국문제목, 국문요약과 국문키워드, 영문제목, 영문요약과 영문키워드, 본문, 감사의 글(필요 시), 참고문헌을 순서대로 포함한다.

1) 영문제목
제목은 공백을 포함해 길이가 40자를 초과하지 않도록 한다.
2) 영문요약과 키워드
요약은 무슨 연구를 어떻게 수행하였는지, 주된 연구결과와 그 중요성에 관해 간결하게 기술하여야 한다. 요약은 300단어를 초과해서는 안되며, 표나 참고문헌 번호를 포함하지 않은 하나의 문단으로 기술되어야 한다. 초록의 하단부에는 연구분야와 내용을 나타낼 수 있는 3 ~ 5단어 이내의 키워드를 기재하여야 한다.
3) 본문
a) 장절 제목: 장이나 절의 제목은 1. , 1.1, 1), a) 와 같이 4 단계 레벨로 표기할 수 있다.
b) 본문 중 참고문헌 인용: 참고문헌은 본문에서 처음 인용되는 순서대로 번호를 붙인다. 그리고 본문에서 참고문헌을 인용할 때는 해당 참고문헌의 번호를 [1, 4, 7] 혹은 [6-9]와 같이 각괄호 안에 기재한다.
c) 약어: 약어는 저자의 편의성보다는 독자에게 도움을 줄 수 있는 방식으로 사용되어야 한다. 따라서 약어는 가급적 제한적으로 사용하는 것이 바람직하다. 표와 그림을 포함해 본문에서 세 번 이상 등장하지 않는 약어의 사용은 가급적 파하라. 약어는 본문에서 처음 사용될 때 축약 이전의 형태로 정의되어야 한다.
d) 표: 표는 본문에서 인용되는 순서대로 아라비아 숫자로 번호를 붙인다. 표의 제목과 설명은 영어로 작성하며, 본문 내용을 읽지 않고도 이해할 수 있도록 간결 명료하게 작성한다
e) 그림: 그림은 본문에서 인용되는 순서대로 아라비아 숫자로 번호를 붙인다. 동일한 번호에 두 개 이상의 그림이 있는 경우, Fig. 1A, Fig. 1B와 같이 아라비아 숫자 뒤에 알파벳 대문자를 기입하여 구분한다. 자신이 그린 그림이 아니면 저작권자의 허락을 받아야 하며 각주에 이를 밝혀야 한다.
4) 감사의 글
필요한 경우, 본문 뒤에 감사의 글을 포함시킬 수 있으며, 연구비 지원 또는 다른 지원에 대한 내용을 명시할 수 있다.
5) 참고문헌
모든 참고문헌은 영어로 기술하며, 제출 원고의 내용과 분명히 관련이 있는 것들이어야 한다. 참고문헌은 본문에서 처음 인용되는 순서대로 번호를 붙인다. 참고문헌들은 반드시 원저 확인을 통해 출처를 검증하는 것이 필요하다.
다음 예시들을 참고하여 참고문헌들을 작성한다.

Journal Article

[1] S. Y. Hea and E. G. Kim, "Design and implementation of the differential contents organization system based on each learner's level," *The KIPS Transactions: Part A*, vol. 18, no. 6, pp. 19-31, 2011.

[2] S. Y. Hea, E. G, Kim, and G. D. Hong, "Design and implementation of the differential contents organization system based on each learner's level," *KIPS Transactions on Software and Data Engineering*, vol. 19, no. 3, pp. 19-31, 2012.

Book & Book Chapter

[3] S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*," 3th ed., New York: Prentice Hall, 2009.

[4] J. L. Hennessy and D. A. Patterson, "Instruction-level parallelism and its exploitation," in *Computer Architecture: A Quantitative Approach*, 4th ed., San Francisco, CA: Morgan Kaufmann Pub., ch. 2, pp. 66-153, 2007.

[5] D. B. Lenat, " Programming articitical interlligence," in *Understanding Artificial Intelligence*, Scientific American, Ed., New York: Warner Books Inc., pp. 23-29, 2002.

Conference Proceedings

[6] A. Stoffel, D. Spretke, H. Kinnemann, and D. A. Keim, "Enhancing document structure analysis using visual analytics," in *Proceedings of the ACM Symposium on Applied Computing*, Sierre, 2010, pp. 8-12.

Dissertations

[7] J. Y. Seo, " Text driven construction of discourse structures for understanding descriptive texts," Ph.D. dissertation, University of Texas at Austin, TX, USA, 1990.

Online Source

[8] Thomas Claburn, Google Chrome 18 brings faster graphics [Internet], http://www.techweb.com/news/232800057/google-chrome-18-brings-faster-graphics.html.

6. 투고료 및 게재료

6.1 투고료

본 학술지에 원고를 투고할 때, 투고자는 1편당 일반 심사의 경우 50,000원(US $50), 급행 심사의 경우 350,000원(US $350)을 학회에 납부하여야 한다.

6.2 게재료

채택된 논문의 투고자는 논문의 게재를 위해 다음과 같은 논문 게재료를 학회 사무국에 납부하여야 한다.
- 인쇄쪽수가 1 ~ 6쪽인 경우, 100,000원
- 인쇄쪽수가 7쪽 이상인 경우, 100,000원 + 50,000원 추가 / 쪽당

6.3 은행계좌

- 한국외환은행: 232-13-01249-5 (예금주: 한국정보처리학회)
- 우체국: 012559-01-000730 (예금주: 한국정보처리학회)

7. 본 투고 규정은 2012년 9월 1일부터 효력을 발생한다.

입회 안내

국가가 지향하는 첨단 정보처리 산업과 기술혁신의 시대에 부응하여 첫째, 정보처리 학술활동의 활성화, 둘째, 정보처리 기술의 산학연 협동의 내실화, 셋째, 정보처리 기술의 국제화와 표준화 등 회원봉사 활동에 역점을 두고 정보화사회를 선도하는 명실상부한 정보처리 분야의 정통학회인 사단법인 한국정보처리학회에서는 정보처리분야에 종사하고 계시는 여러분들의 많은 입회를 바라고 있습니다.

주 요 목 적 사 업

1. 정보처리에 관한 각종 학술발표회 및 전시회 개최
2. 정보처리에 관한 지식 및 기술 보급에 관한 사업
3. 정보처리 기술의 상호 협조 및 정보 교환
4. 정보처리에 관한 표준화 사업
5. 국제적 학술 교류 및 기술 협력
6. 학회지 및 논문지 발간
7. 정보처리에 관한 문헌 발간
8. 기타 본 학회 목적 달성에 필요한 사업

(정관 제4조)

회 원 의 종 류 및 자 격

1. 특별회원 : 정보처리 분야 발전에 기여하고 본학회의 취지에 찬동하는 법인 및 단체.
2. 명예회원 : 학식과 덕망이 높고 본 학회의 발전에 크게 기여한 자.
3. 정 회 원 : 정보처리 관련 분야를 전공하여 학사학위 이상을 취득한 자 또는 정보처리 관련분야에서 2년이상 근무한 자.
4. 준 회 원 : 정보처리 관련학과 학생 또는 대학원생
5. 단체회원 : 도서관 또는 초·중·고 교육기관

(정관 제6조)

회 원 의 혜 택

1. 정보처리학회지(논설, 기술보고, 해설, 전망, 강좌, 단편정보 등 게재) 발행. 무료배포
2. 정보처리학회논문지 및 특집호(학술연구논문, 심사완료 후 게재) 발행.
3. 춘추계 학술발표회와 각종 학술행사에 참가 및 논문발표
4. 전문분과연구회의 활동자격과 각종 학술행사에 참가 및 논문발표
5. 국제 학술회의 활동 및 외국 학회에 참가 및 추천
6. 정보처리 및 기술발전에 업적이 있는 회원에게는 각종 학회상 수여

회 비

1. 특별회원 회비는 이사회의 결정에 따르면 종신회원·정회원·준회원·단체회원 회비는 다음과 같다.

구 분	종 신 회 원	정 회 원	준 회 원	단 체 회 원
연 회 비	600,000원	60,000원	40,000원	300,000원

※ 논문 구독료 각권 별도 2만원 (필요시 구독)

2. 회원가입은 학회 홈페이지를 통하여 회원정보를 입력하신후 회비를 신용카드 결제 및 아래의 은행으로 입금하여 주시기 바랍니다.
   한국외환은행 계좌번호 : 232-13-01249-5 예금주 : (사단)한국정보처리학회
   우 체 국 계좌번호 : 012559-01-000730 예금주 : 한국정보처리학회

문의처 : 140-750 서울특별시 용산구 한강대로 109, 1002호(한강로 2가 용성비즈텔)
사단법인 한국정보처리학회 사무국 귀하
전 화 : (02) 2077-1414(代) 팩 스 : (02) 2077-1472
홈페이지 : www.kips.or.kr e-mail : ysyun@kips.or.kr

연구회 안내

당 학회에는 현재 다음과 같은 연구회가 구성되어 있으며, 이들 연구회는 위원장을 중심으로 하여 현재 활발한 연구 활동을 하고 있습니다. 연구회에 가입을 원하시는 회원은 연구회 가입 원서를 작성하셔서 당 학회 사무국 또는 각 위원장에게 보내주시기 바랍니다. 회원 여러분의 많은 가입을 부탁드립니다. 연구회 발족 등에 관한 의견이 있으시면 학회로 연락 주시기 바랍니다.

e-Bridge 연구회

위원장 : 이정배 부총장 (부산외국어대학교)
전 화 : 051)509-5033
e-mail : jblee1120@naver.com

우정기술연구회

위원장 : 정 훈 부장 (ETRI)
전 화 : 042)860-6470
e-mail : hoonjung@etri.re.kr

IT 융합서비스연구회

위원장 : 박석천 교수 (가천대학교)
전 화 : 031)750-5328
e-mail : scpark@gachon.ac.kr

전산교육연구회

위원장 : 임관철 교수 (대전보건대학교)
전 화 : 042-670-9354
e-mail : kcim@hit.ac.kr

IT 정책연구회

위원장 : 오길록 교수 (숭실대학교)
전 화 :
e-mail : gilrokoh@paran.com

전산수학연구회

위원장 : 박진홍 교수 (선문대학교)
전 화 : 041)530-2224
e-mail : chp@omega.sunmoon.ac.kr

빅데이터컴퓨팅연구회

위원장 : 이필규 교수 (인하대학교)
전 화 : 032)860-7448
e-mail : pkrhee@inha.ac.kr

전자정부연구회

위원장 : 이재두 수석 (NIA)
전 화 : 02)2131-0370
e-mail : leejaedu@gmail.com

소프트웨어공학연구회

위원장 : 박용범 교수 (단국대학교)
전 화 : 041)550-3464
e-mail : ybpark@dankook.ac.kr

정보통신융용연구회

위원장 : 오진태 부장 (ETRI)
전 화 : 042)860-4977
e-mail : showme@etri.re.kr

스토리지시스템연구회

위원장 : 신범주 교수 (부산대학교)
전 화 : 055-350-5417
e-mail : bjshin@pusan.ac.kr

지식및데이터공학연구회

위원장 : 진병운 박사 (ETRI)
전 화 : 042)860-6544
e-mail : bwjin@etri.re.kr

에너지그리드정보처리연구회

위원장 : 이봉재 센터장 (전력연구원)
전 화 : 042-865-5700
e-mail : leeboja@kepco.kco.kr

컴퓨터소프트웨어연구회

위원장 : 박두순 교수 (순천향대학교)
전 화 : 041)530-1317
e-mail : parkds@sch.ac.kr

한국정보처리학회 신용카드 결제신청서

◈ **납입방법 : 신용카드**

◈ **결재내용 : 학회 회비 / 세미나 참가비 / 논문 구독료 / 논문 게재료**

학 회 회 비	종신회원 ₩600,000( )	정회원 ₩60,000( )
	준 회 원 ₩40,000( )	기 타 (₩)
행 사 등 록 비	(₩)	
논 문 구 독 료 (각 권당 2만원)	□소프트웨어 및 데이터 공학(KTSDE) □컴퓨터 및 통신 시스템(KTCCS) (₩)	
논 문 게 재 료	( )권 ( )호 (₩)	
기 타	(₩)	

◈ **신용카드 사용내역서**

카 드 명	□ 신한카드 □ 국민카드 □ 비씨카드		결 재	일시불( )	※ 타카드 사용 불가
카드번호	□□□□ □□□□ □□□□ □□□□				
지불금액	원	카드유효기간	년 월	전 화	
소 속		성 명		서 명	
"상기 금액을 정히 지불합니다" 사단법인 한국정보처리학회					

※ 신한카드, 국민카드 및 비씨카드만 사용이 가능합니다.

※ 반드시 팩스로 회송바랍니다.

※ 학회 연회비 및 논문 구독료는 홈페이지에서 로그인 후 모든 카드로 온라인 카드 결재가 가능합니다.

☞ 보내실곳 : **한국정보처리학회**

전화 : (02)2077-1414 팩 스 : (02)2077-1472

http://www.kips.or.kr e-mail : ysyun@kips.or.kr

140-750 서울특별시 용산구 한강대로 109, 1002호(한강로 2가 용성비즈텔)

학 회 사 무 국

선임국장	송영민 (내선 5)	min@kips.or.kr	업무총괄 / 제회 / CUTE 행사 / SQMS 행사
국 장	김은순 (내선 2)	uskim@kips.or.kr	학회지 / 춘계학술대회 / 단기강좌 / 연구과제
과 장	이주연 (내선 1)	joo@kips.or.kr	JIPS(영문지) / IT21컨퍼런스 / 추계학술발표대회
대 리	윤영숙 (내선 3)	ysyun@kips.or.kr	회원 / 재무 / 국문지 / 홈페이지 및뉴스레터

- **사무국주소** : (04376) 서울특별시 용산구 한강대로 109, 1002호(한강로2가 용성비즈텔)
- **전 화** : 02) 2077-1414
- **팩 스** : 02) 2077-1472
- **대 표 메 일** : kips@kips.or.kr
- **홈 페 이 지** : www.kips.or.kr

정보처리학회지

제 23 권 제 4 호

등록일자 : 1994년 3월 31일
서기 2016년 7월 25일 인쇄
서기 2016년 7월 30일 발행

발 행 인 : 구 원 모

편 집 인 : 강 윤 희, 최 민

발 행 처 : 사단법인 한국정보처리학회 KIPS Korea Information Processing Society

(140-750) 서울특별시 용산구 한강대로 109, 1002호(한강로 2가 용성비즈텔)
전 화 : (02)2077-1414(代) 팩 스 : (02)2077-1472
홈페이지 : www.kips.or.kr 이메일 : kips@kips.or.kr

* 제작 : (주)이환디앤비 Tel : (02)2254-4301(代) <비매품>

THE KIPS FALL CONFERENCE 2016
https://manuscriptlink.com/society/kips/conference/kips2016fall

2016년 추계학술발표대회

일자 | 2016년 11월 4일(금) ~ 5일(토) 장소 | 부산대학교(부산캠퍼스)

논문모집개요

논문모집내용
- 정보처리 분야의 학술 논문
- 개발 성공 사례 및 개발 완료 또는 개발 중인 연구 과제

논문발표분야 소프트웨어공학, 데이터베이스, 사물인터넷(IoT), 멀티미디어처리, 인간컴퓨터상호작용, 인공지능, 통신시스템, 정보보호, 모바일컴퓨팅, 컴퓨터교육, 병렬 및 분산컴퓨팅, 컴퓨터시스템 및 이론, 인터넷응용, 정보시스템

논문작성방법 행사 홈페이지에서 제공하는 논문양식을 다운받아 **구두 논문은 3~4페이지, 포스터 논문은 2페이지로 작성** (발표유형에 따른 작성 제한 페이지 반드시 엄수)

논문제출방법 홈페이지 퀵 메뉴 추계학술발표대회에서 로그인 후 제출 **(학술대회 논문 제출자는 학회 회원 가입(또는 기존 회원 인증) 이후 회비 납부 유무와 관계없이 언제나 논문 제출이 가능합니다. 다만, 추후 학술대회 사전등록시 반드시 학회 회비 및 참가비를 함께 납부해주셔야 합니다.)** 모든 안내는 **제출시 입력하신 이메일, 연락처** 등으로 공지가 되오니, **정보를 정확히 입력**하여 주시기 바랍니다.

행사주요일정

논문제출마감	2016년 **9월 26일(월)**	자정까지(최종마감)
심사결과공지	2016년 10월 5일(수)	**홈페이지 공고**
최종논문제출	2016년 10월 10일(월)	자정까지
발표자사전등록	2016년 10월 10일(월)	자정까지 (논문은 편당 등록)

시상 및 참고사항

- 발표한 논문 중 최우수논문 및 우수논문을 선정하여 총회에서 시상
- **최우수논문은** JIPS(SCOPUS, 등재지)에 추천하고, **우수논문은** 한국정보처리학회 논문지(등재지)에 추천 예정(행사종료 후 메일 안내)
- 학부생들의 참신한 아이디어를 발굴하고 발표 능력을 향상시키기 위한 목적으로 **학부생 논문은 별도로 심사 예정**(학부생 캡스톤 경진대회 결과물도 제출 가능)

사단법인 KIPS 한국정보처리학회 Korea Information Processing Society | 행사(논문 및 진행) 문의 학회 사무국, 이주연 과장 Tel. 02-2077-1414(내선 1) E-Mail. joo@kips.or.kr Web. www.kips.or.kr

LG히다찌
http://www.lghitachi.co.kr

LG히다찌가
Smart 세상을 만들어 갑니다