

계산과학분야의 고성능컴퓨팅 교육 개선을 위한 탐색적 연구

윤희준¹, 안성진^{2*}

¹성균관대학교 교과교육학과 박사과정, ²성균관대학교 컴퓨터교육과 교수

A Study on the Improvement of High Performance Computing Education in Computational Science

Heejun Yoon¹, Seongjin Ahn^{2*}

¹Department of Disciplinary Education, Sungkyunkwan University, Ph. D. student

²Department of Computer Education, Sungkyunkwan University, Professor

요 약 계산과학분야에서 고성능컴퓨팅(HPC)을 활용하기 위해서는 프로그래밍, 알고리즘, 자료구조 등 컴퓨터과학의 지식들과 기술들을 배워야 한다. 본 논문에서는 계산과학분야의 IT교육현황 조사와 설문조사를 통해 고성능컴퓨팅 교육을 개선시키기 위한 정책 방향을 제안하는데 있다. 이를 위해 국내 대학의 물리학, 화학, 생명과학, 지구과학분야의 전공과목 중에서 IT관련 과목 현황과 사용자들의 국내 고성능컴퓨팅 교육에 대한 인식을 조사하였다. 그 결과 계산과학분야의 IT과목비율은 응용 전공과목에 비해 매우 낮았다. 대학의 교육 요구도는 높게 나왔지만, 대학의 교육 제공 수준은 제일 낮게 나왔다. 또한 대부분의 사용자들은 독학으로 필요한 지식과 기술들을 습득한 것으로 조사되었다. 즉 대학의 역할이 가장 시급하고 중요하며 전문기관과 온라인교육의 역할도 중요하다고 확인하였다.

주제어 : 고성능컴퓨팅, 고성능컴퓨팅 활용 교육, 고성능컴퓨팅 교육과정, 계산과학, 교육 정책

Abstract In order to utilize HPC in Computational science, It is necessary to learn the knowledge and skills of computer science such as programming, algorithms and data structure. In this paper, we investigate IT education status in Computational science and propose policy directions to improve the HPC education through user survey. To do this, we surveyed the current state of IT subjects among major subjects in physics, chemistry, life sciences, and earth science in domestic universities and surveyed the users' Recognition of HPC education. As a result, the ratio of IT subjects in Computational science was very lower than the ratio of major domain subjects. Despite the high educational needs of universities, the educational level of universities was the lowest. Most users have learned the necessary knowledge and skills through self-study. We recognized the role of the university is the most urgent and important, and the role of professional institutions and online education is also important.

Key Words : HPC, HPC Education, HPC Curriculum, Education Policy, Computational science

1. 서론

고성능컴퓨팅(High Performance Computing, HPC)은 지난 몇 십 년 동안 물리학, 화학, 생명과학, 지구과학 등

다양한 기초 과학 연구 분야에서 거대 문제 계산을 위해 주로 활용되어 왔다[1]. 또한 최근에는 여러 선진국에서 국가안보, 지구환경, 신약개발, 자원탐사 등 다양한 분야에서 국가 현안 해결과 기업 경쟁력 강화를 위해 활용

*Corresponding Author : Seongjin Ahn(sjahn@skku.edu)

Received September 5, 2018

Accepted December 20, 2018

Revised October 26, 2018

Published December 28, 2018

[2,3]하고 있다. 이는 고성능컴퓨팅의 성능이 무어의 법칙 [4]을 넘어 빠르게 발전하면서 과학적인 성과와 산업, 경제 등 국가의 모두 분야에 미래 경쟁력을 좌우하는 중요한 요소로 인식되고 있기 때문이다. 특히 오랫동안 고성능컴퓨팅을 사용해온 계산과학은 컴퓨터의 계산을 이용하는 모의실험을 통해 연구 문제를 해결하는 학문으로 주로 과학영역, 수학영역, 컴퓨터과학 영역으로 융합된 학문 분야이다[5]. 계산과학분야에서 고성능컴퓨팅을 활용하기 위해서는 컴퓨터 아키텍처, 알고리즘, 프로그래밍 등 컴퓨터과학의 기본적인 지식들과 기술들을 필요로 한다[6,7]. 또한 빠른 속도의 계산과 대용량의 데이터를 처리하기 위해 병렬 프로그래밍, 대용량데이터 처리, GPU 기반 프로그래밍, 가상화 기술 등 최신 주제와 다양한 기술들[8,9]도 필요로 하고 있다.

이에 국내에서는 2011년부터 고성능컴퓨팅 수요에 따라 초고성능컴퓨터 활용 및 육성에 관한 법률[10]을 제정하여 고성능컴퓨터 하드웨어의 개발, 고성능컴퓨터 운영, 고성능컴퓨터 기반의 응용 소프트웨어 개발 및 응용 등 고성능컴퓨팅을 활용하여 다양한 연구개발을 잘 수행할 수 있도록 노력하고 있다. 하지만 고성능컴퓨터를 개발할 수 있는 하드웨어 원천기술과 이를 활용하는 소프트웨어 개발과 활용, 거대한 응용연구, 고성능컴퓨팅 활용 인력 양성 등 많은 영역에서 선진국에 비해 많이 부족한 상황이다. 또한 지능정보사회로 진입함에 따라 고성능컴퓨팅을 활용하는 분야가 지속적으로 확장[11,12]되고 있으며 이에 따라 고성능컴퓨팅 활용 인력에 대한 수요가 급증하고 있지만 고성능컴퓨팅 활용 인력의 공급은 이에 미치지 못하고 있다. 소수의 대학과 전문기관에서 고성능컴퓨팅 활용에 대한 교육을 제공하고 있지만 수요를 충족시키지 못하고 있다. 보다 근본적인 문제는 고성능컴퓨팅 활용 인력을 양성하기 위한 체계적인 교육체제가 갖춰지지 않아 대학 졸업생이나 단기 기초교육 이수자들이 현장에서 필요로 하는 실무적인 역량 즉, 지식과 기술들을 확보하지 못하고 있는 것이다.

본 연구는 국내 계산과학분야 중심으로 IT 교육현황을 조사하고 사용자들의 인식을 조사하여 현실적인 요인을 반영한 고성능컴퓨팅 활용을 위한 정책적인 교육 방향을 제안하는데 목적을 두고 있다. 연구의 범위는 고성능컴퓨팅을 활용하는 자연과학대학의 물리학, 화학, 생명과학, 지구과학 분야로 한정하였으며 전공과목 중에서 IT 관련 과목을 조사하고 해당분야에서 고성능컴퓨팅을

활용하는 사용자들을 대상으로 설문 조사한 결과를 통해 시사점과 체계적인 교육과정을 설계하고 개발하기 위한 개선방향을 도출하고자 한다. 연구에서 제시하는 고성능컴퓨팅 교육 방향은 계산과학분야뿐만 아니라 고성능컴퓨팅을 활용하는 다른 분야와 융합되어 교육 정책과 의사결정에 참고할 수 있는 역할을 할 수 있도록 하였다.

2. 국내외 고성능컴퓨팅 활용 교육 현황

2.1 고성능컴퓨팅 교육 관련 연구

Table 1은 1고성능컴퓨팅 교육에 관한 연구들을 간략히 보여주고 있다. 2000년대 초반까지는 컴퓨터과학의 프로그래밍, 컴퓨터구조와 같은 기본적인 지식과 기술들에 대한 교육 프로그램을 제공하고[1,5,13] 있지만 2010년 전후부터 컴퓨터과학의 기본적인 지식과 기술뿐만 아니라 병렬 프로그래밍, 병렬 알고리즘, 병렬 컴퓨팅 등 세분화된 병렬에 관한 지식과 기술들에 대한 교육을 제공 [14-17]하고 있다.

Table 1. Researches on HPC Education Programs

Researchers	Educational Programs
Gallopolous, S., Houstis, E. N., & Rice, J. R. (1992)	Object-oriented techniques, Software Interconnection Technologies, Language and Compiler Technology, Parallel and Distributed Computation, Networks
Stevenson, D. E. (1993, March)	Architecture, operation systems, graphics, database management, software engineering, programming language
Yasar, O., & Landau, R. H. (2003)	programming, performance benchmarks, relationship among architecture/language/performance, parallel libraries
Fabricius, U., Freundl, C., Köstler, H., & Rüde, U. (2005)	Algorithms, Data Structure, Parallel computing, Parallel Algorithms, Programming Techniques, Computer architecture, code profiling, tuning, optimization, cache blocking
S. W. G. O. C. U., P. T. A. L. P., Shiflet, A., Vakalis, I., Jordan, K., & John, S. S. (2011)	programming, algorithms, a high-level language, data structures and analysis, parallel programming
Gergel, V., Meyerov, I., & Sysoyev, A. (2015)	Programming, Data Structures, Algorithms, Computer Architecture, Parallel Computing, parallelism, parallel programming, application of tools, parallel algorithms
EUROLAB-4-HPC (2017)	Parallel Computer Architectures, Scalable parallel algorithms, Programming with MPI, Data Parallel Computing, Programming Shared Memory Parallel Systems, Programming Multi-core and Many-core Systems, Performance Engineering, Programming Heterogeneous and Accelerated Systems, Large scale Scientific Computation, Data Science Fundamentals

2.2 미국 고성능컴퓨팅 교육

미국은 1990년대에 Michigan대학, Rice대학, Stanford대학 등 6개의 대학에서 계산과학분야의 전문 교육과정을 제공하고 있었으며[13] 2000년대에는 세분화되어 Illinois State대학, Stanford대학, Carnegie Mellon대학, Michigan State대학, Rice대학 등 22개의 대학에서 계산과학 학부 교육과정과 21개의 부정공 교육과정을 제공하였음[18]. 최근에는 Capital University, Doane College, Florida State University, George Mason University, Ohio State University, Stanford University, University of Minnesota, The University of Texas at Austin, University of Iowa, San Diego State University, Indiana University, The University of Utah 등 많은 대학들이 계산과학 전공자들에게 학부 교육 프로그램과 대학원 과정 교육 프로그램을 제공하고 있다[19]. O. Ya, sar & R. H. Landau(2003)는 컴퓨터과학(CS), 계산과학(CSE), 계산물리(CP), 물리(PH)의 학부 프로그램의 교육과정에 대해 Computing과목, Mathematics과목, Application과목, Other과목의 비율을 조사한 결과는 Table 2와 같다. 계산과학과 계산물리의 컴퓨팅 과목은 20%이었지만 순수물리는 2%로 아주 적게 조사되었다. 또한 Application과목의 비중이 계산과학과 계산물리는 28%로 비슷하였으며 순수물리는 45%로 높게 조사되었다[5].

Table 2. Subject Balance(%Courses) for B.S degree programs in CS(%), CSE(%), CP(%), and PH(%)

	CS	CSE	CP	PH
Other	31	29	32	36
Application	17	28	28	45
Mathematics	12	23	19	17
Computing	40	20	20	2

미국은 전략적으로 대학 자체의 교육뿐만 아니라 국가슈퍼컴퓨팅센터와 연계된 고성능컴퓨팅 교육에 대한 다양한 프로그램을 제공하고 있다. NSF XSEDE(The Extreme Science and Engineering Discovery Environment) 프로젝트를 통해 대학(원)생, 연구자, 관련 시스템 사용자들을 대상으로 국가 차원의 고성능컴퓨팅 교육 및 인력 양성을 하고 있다. NCSA(National Center for Supercomputing Applications, University of Illinois), SDSC(San Diego Supercomputer Center), NICS(National Institute for

Computational Sciences), TACC(Texas Advanced Computing Center), PSC(Pittsburgh Supercomputing Center) 등 여러 국가 슈퍼컴퓨팅센터들과 Cornell대학, Indiana대학, PURDUE대학 등 여러 대학들이 협력하여 교육 프로그램에 참여하고 있다[20,21]. 예를 들면 Texas 대학의 TACC 교육과정은 Scientific Programming, Scientific/Technical Computing, Parallel Computing, Visualization & Data Analysis, HPC with GPUs 로 다양하게 구성되어 있다[22]. 또한 온라인으로 가상의 HPC University를 통해 체계적이며 사용자 수준을 고려하여 프로그래밍과 알고리즘, 최적화, 병렬프로그래밍, 가시화 등 교육 주제와 훈련 방법을 연구자와 학생들에게 제공하고 있다[23,24].

2.3 유럽 고성능컴퓨팅 교육

유럽은 2012년부터 개별국가가 아닌 전 유럽의 연구기관이 협력하여 PRACE(Partnership for Advanced Computing in Europe : 유럽 첨단컴퓨팅 협력체)라는 프로젝트를 통해 고성능컴퓨팅 교육 프로그램을 제공하고 있다. 이 프로젝트는 독일, 영국, 프랑스, 이탈리아, 스위스, 스페인, 네덜란드 등 26개국이 참여[25]하여 페타플롭 시스템, 병렬 프로그래밍, 빅데이터 분석, HPC Tools, GPU프로그래밍, HPC코드 최적화 등 교육 프로그램을 운영하고 있다[26]. 네덜란드의 Amsterdam대학, 영국의 Manchester대학, 스위스의 Lausanne공과대학과 Zurich ETH, 스웨덴의 KTH과 Uppsala대학, 독일의 RWTH Aachen, TU Darmshtadt, TU Munchen, Erlangen대학, 스페인의 UPC/BSC, 핀란드의 Helsinki대학들이 계산과학 교육과정을 제공하고 있다[27]. 계절학과 6개의 PRACE Advanced Training Centres (PATCs), 4개의 PRACE Training Centers(PTCs)에서 다양한 교육 프로그램을 제공하고 있다. 예를 들면 고급훈련센터(PATCs)에서는 병렬 프로그래밍, 병렬시스템, CUPA 프로그래밍 등을 제공하고 영국의 EPCC (The University of Edinburgh)은 대학생, 연구자들을 대상으로 온라인 교육과 고성능컴퓨팅 파트너십 박사과정을 운영하고 있다[28]. 또한 독일, 스페인 등 일부 국가에서는 독립적인 개별 교육 프로그램을 실시하고 있다[29,30].

2.4 국내 고성능컴퓨팅 교육

국내에서 IT 비전공자들을 위한 다양한 소프트웨어교

육[31,32]은 연구되고 있으나 고성능컴퓨팅 교육에 관한 연구는 매우 부족한 상황이다. 미국이나 유럽처럼 많은 대학들이 아닌 소수의 대학과 공공기관에서 일부 제공하고 있다. 서울대학교는 계산과학 전공을 개설하여 학부 과정, 석박사 과정을 통해 고성능컴퓨팅을 활용할 수 있는 인력을 양성하고 있다. 또한 연세대학교에서 전산유체역학, 고급수치해석, 의학영상처리 연구 분야와 관련하여 계산과학 석박사 과정으로 대학원 교육프로그램을 제공하고 있다. 서울대학교와 연세대학교의 계산과학 전공 과정 IT 관련 과목현황은 Table 3과 같다. 그리고 국가슈퍼컴퓨팅센터인 한국과학기술정보연구원에서는 정기교육, 비정기교육, 계절학교, 온라인 및 집합교육 프로그램을 제공하고 있다. 이를 통해 리눅스시스템, 슈퍼컴퓨터시스템 등 시스템, GPU, MPI 등 병렬 컴퓨팅 및 병렬 프로그램, Python, Fortran 등 프로그래밍언어, 수치해석 라이브러리 활용, 최적화, 디버깅 등 고성능컴퓨팅과 관련된 교육을 실시하고 있다[33]. EDISION을 통해 e-learning할 수 있는 교육 내용이 온라인으로 제공되고 있다. 전산열유체, 나노물리, 계산물리, 전산의학 등 여러 분야의 주제들로 구성되어 있으며 전산유체역학개론, 공간차분법 및 차분정확도, 수리물리학, 3차원날개 해석, 난류이론 등 온라인 강의도 제공하고 있다[34].

Table 3. Researches on HPC Education Programs

University	IT Course
Seoul University	Theory and Practice in Computational Sciences, Data Introduction to Scientific Computing, Algorithms, Data Structures, Computer Animation
Yonsei University	Basics of Computational Science and Engineering, Medical Imaging System, Programming for Image Processing

3. 국내 자연과학대학 IT교육 현황 및 고성능컴퓨팅 인식 조사

3.1 국내 자연과학대학 교육과정 조사

국내 대학은 계산과학 전공학자가 많지 않아 계산과학분야인 물리학, 화학, 생명과학, 지구과학의 전공교육 현황에 대한 조사를 다음과 같이 수행하였다. 조사내용은 각 학과의 전공과목 현황과 전공과목 중에서 IT과목들을 조사한 후 각 분야 학과의 석사과정 1명과 박사 1명

에게 과목내용에 대한 검증을 받았다. 그리고 검증받은 과목들을 기준으로 각 분야별 전공과목과 IT과목의 비율을 조사하였다.

대학 알리미의 표준분류체계에 따라 국내 4년제 대학을 중심으로 2017년 국내 자연과학의 물리학, 화학, 생명과학, 지구·천문·기상학과가 있는 대학과 학과를 대상 [35]으로 2018년 3월 25일부터 2018년 4월3일까지 조사를 하였다. 조사한 대학은 강원원주대, 강원대, 건국대, 경기대, 경북대, 경상대, 경성대, 경희대, 계명대, 고려대, 공주대, 광운대, 군산대, 단국대, 대구대, 덕성여대, 동국대, 동아대, 명지대, 목포대, 부경대, 부산대, 상지대, 서강대, 서경대, 서울대, 서울시립대, 서울여자대, 성균관대, 성신여대, 세종대, 숙명여대, 순천향대, 숭실대, 아주대, 안동대, 연세대, 영남대, 용인대, 울산대, 원광대, 이화여대, 인천대, 인하대, 전남대, 제주대, 조선대, 중앙대, 창원대, 청주대, 충남대, 충북대, 한국과학기술원, 한국외대, 한남대, 한림대, 한양대, 협성대, 호서대로 60개 대학이며 학과는 교육과정 정보가 없는 1개 학과를 제외한 152학과이다. 세부분야는 물리학과 39개, 화학과 49개, 생명과학과 48개, 지구과학과 16개이다. 조사된 자연과학분야의 IT과목들은 Table 4와 같다.

Table 4. IT Subjects in Computational Science

Field	IT Subjects
Physics	Computational Physics, Computational Physics and Practice, Advanced Applications Computational Physics, Computer Physics and Practice, Information Physics, Computational Display Physics, Computational Engineering Physics, Basic Computational Physics, IT convergence measurement programming, Programming and physics, Advanced Computational Physics, Introduction to computer language, Computer programming, Computer simulation, Basic programming language
Chemistry	Introduction to Computational Chemistry and Practice, Computational Chemistry, Computer Chemistry, Computers and Modern Chemistry, Advanced Computational chemistry, Computer Applications Chemistry, Chemical Computation and Practice, Chemistry and software, Chemistry Informatics and Practice, Chemistry Informatics, Chemical computation, Computer Chemistry and Practice, Computational Mathematics, STEM numerical programming, Introduction to Chemical Information, Computational linear algebra, Chemical simulation
Bioinformatics	BioPython, Bioinformatics, Bioinformation engineering, Medical Informatics, Bioinformatics and Proteomics, Introduction to Bioinformatics, Introduction to Computer Science for Biologists, Programming Language and Practice

Earth Atmospheric Sciences	Fortran programming for Atmospheric Sciences, Atmospheric Environmental Information System, Earth Informatics and Practice, Computer Programming for Atmospheric Science and Practice, Computational Astronomy and Practice, Astronomical Image and Data Processing and Practice, Space and computers, Data Analysis in Atmospheric Sciences, Scientific Computing & Programming in Earth Sciences, Computer Earth Systems Science and Practice, Computational Fluid Dynamics, Computational Numerical Analysis and Practice, Basic programming and practice, Advanced object-oriented programming, Programming and Practice
----------------------------	--

Fig. 1은 IT과목들이 포함된 학과수를 보여주고 있다. ‘전공교육과목 중에서 컴퓨터과학 과목이 1과목 이상’ 포함되어 있는 학과 비율은 물리학이 89.7%, 화학이 57.1%, 생명과학이 54.2%, 지구과학이 75%로 나왔으며 ‘전공교육과목 중에서 전산과목이 2과목 이상’ 포함되어 있는 학과 비율은 물리학이 21.3%, 화학이 12.2%, 생명과학이 6.3%, 지구과학이 37.5%로 조사되었다.

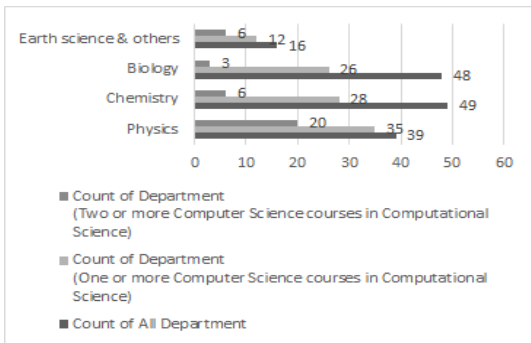


Fig. 1. IT course status in Computational Science

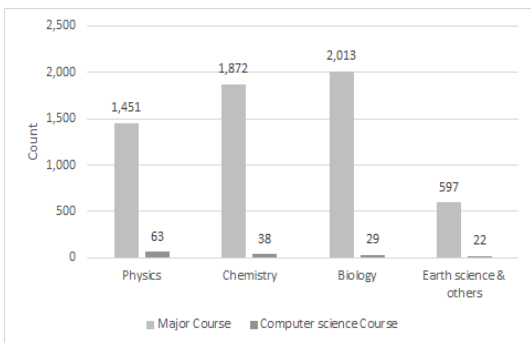


Fig. 2. Major course vs Computer science course in Computational Science

그러나 ‘분야별 전체 전공과목 수와 IT과목 수’에 대

해서는 Fig. 2에서 보여주는 것처럼 물리학이 63과목, 화학이 38과목, 생명과학이 29과목, 지구·천문·우주·대기과학 22과목으로 조사되었다. 계산과학분야의 전공 과목수와 비교할 때 IT과목의 비율은 물리 4.3%, 화학 2%, 생명과학 1.4%, 지구과학 3.7%로 매우 낮게 조사되었다. 각 분야의 학과마다 1~2개의 IT과목만이 포함되어 고성능 컴퓨팅을 활용하기 위한 계산과학분야의 교육과정에 IT과목을 추가하여 균형을 맞출 필요성이 제기된다.

또한 조사된 IT과목들을 검증한 박사들을 대상으로 과목의 내용에 대해 심층 면담하였으며, 결과는 다음과 같다. 먼저 대부분의 IT과목들이 기초 프로그래밍을 다루고 있으며 간단한 코딩을 배운 후 각 분야의 특정 응용 소프트웨어에 적용 및 사용하는 것을 배운다고 공통적인 의견을 주었다. 또한 컴퓨터과학 전공과목의 모든 내용이 필요하지 않고 일부만 필요하다는 것이다. 예를 들면 대부분 C프로그래밍은 배웠지만 알고리즘이나 컴퓨터구조에 대해 심도 있게 배울 필요는 없다고 한다. 또한 차이점은 프로그래밍의 수준이 각각 연구하는 주제에 따라 다르다는 것이다. 일부는 프로그램을 변경하여 사용할 수 있는 수준이면 충분한 반면, 일부는 단순한 변경이 아닌 많은 부분을 프로그램으로 직접 개발해야 하는 경우도 있다고 하였다. 즉 단순한 기초 프로그래밍뿐만 아니라 다양한 IT교육과목이 필요하다

3.2 고성능컴퓨팅 교육에 대한 인식 조사

본 연구에서는 계산과학 전공자들을 대상으로 고성능 컴퓨팅 교육에 관한 설문 도구를 만들었다. 설문도구는 고성능컴퓨팅 교육에 대한 ‘교육의 필요성’, ‘교육 환경의 적절성’, ‘교육 경험’, ‘지식과 기술을 습득한 경로’, ‘대학·전문기관·온라인 등 교육기관별 중요수준’, ‘대학·전문기관·온라인 등 교육기관별 제공수준’의 항목으로 구성되었다. 설문은 학부를 졸업하고 석사과정 이상의 계산과학 전공자들 43명을 대상으로 하였다. 일반적인 통계사항은 다음과 같다. 학위현황은 박사졸업 42%(18명), 석사졸업 30%(13명), 학부졸업 26%(11명), 무응답 2%(1명)이다. 주요 활용하는 분야는 물리 35%(15명), 기계 21%(9명), 화학 및 재료 19%(8명), 지구과학 및 기상 5%(2명), 생명과학 2%(1명), 기타 19%(8명)로 조사되었다. 고성능 컴퓨팅을 활용한 경력은 9년 이상이 14%(6명), 6년~9년 미만이 5%(2명), 3년~6년 미만이 14%(6명), 3년 미만이 67%(29명)이다.

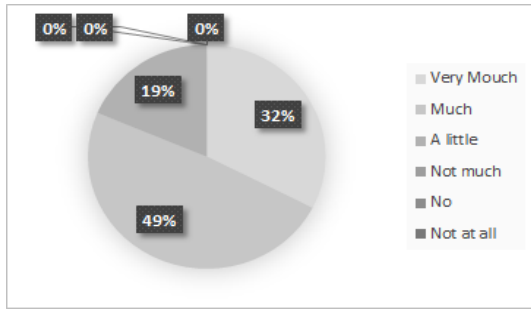


Fig. 3. Need for HPC Education

먼저 Fig. 3과 같이 고성능컴퓨팅 활용에 대한 교육의 필요성에 대해 ‘아주 많이 필요하다’가 14명(32%), ‘많이 필요하다’가 21명(49%), ‘약간 필요하다’가 8명(19%)으로 응답자 모두가 고성능컴퓨팅 활용에 대한 교육이 필요하다고 응답하였으며 ‘많이 필요하다’는 이상의 응답이 35명(81%)로 나타났으며 교육이 ‘필요없다’는 부정적인 의견은 모두 0%를 보여주고 있다.

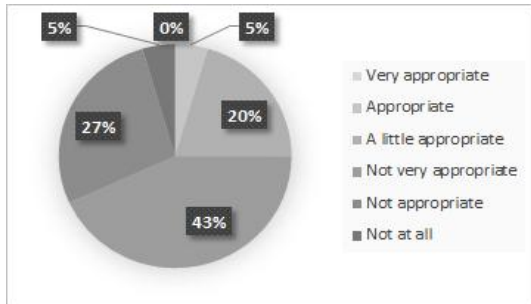


Fig. 4. Environment of HPC education

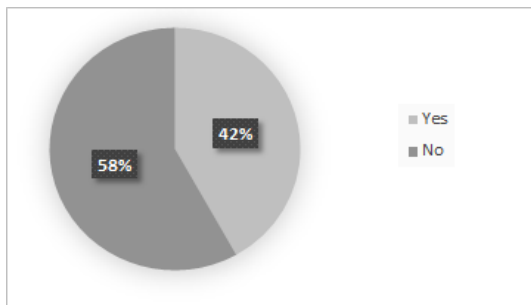


Fig. 5. Experience of HPC education

Fig. 4와 같이 고성능컴퓨팅 활용에 대한 국내 교육환경의 적절성에 대해 ‘매우 적절하다’가 0명(0%), ‘적절하

다’가 2명(5%), ‘약간 적절하다’가 20명(20%), ‘별로 적절하지 않다’가 19명(43%), ‘적절하지 않다’가 12명(27%), ‘전혀 적절하지 않다’가 2명(5%)으로 고성능컴퓨팅 교육 환경에 대해 긍정적인 인식은 25%이며 부정적인 인식은 75%로 조사되었다. Fig. 5와 같이 고성능컴퓨팅 교육을 받은 경험이 있는지에 대해 ‘받은 적이 있다’는 18명(42%), ‘받은 적이 없다’는 25명(58%)로 학부교육과정에서 50%이상이 고성능컴퓨팅 교육을 받지 못한 것으로 조사되었다.

Fig. 6과 같이 본인이 고성능컴퓨팅활용에 대한 지식과 기술을 제일 많이 습득한 경로에 대해 ‘대학’은 3명(7%), ‘전문기관’은 5명(12%), ‘독학’은 22명(51%), ‘지인’은 7명(16%), ‘기타’는 6명(14%)로 나타났다.

Fig. 7과 같이 고성능컴퓨팅 활용에 대한 교육기관 유형별 중요도에 대해 ‘대학이 중요하다’는 37명(86.1%), ‘대학이 중요하지 않다’는 6명(13.9%), ‘전문기관이 중요하다’는 41명(95.4%), ‘전문기관이 중요하지 않다’는 2명(4.6%), ‘온라인 교육이 중요하다’는 33명(76.7%), ‘온라인 교육이 중요하지 않다’는 10명(23.3%)로 나타났다.

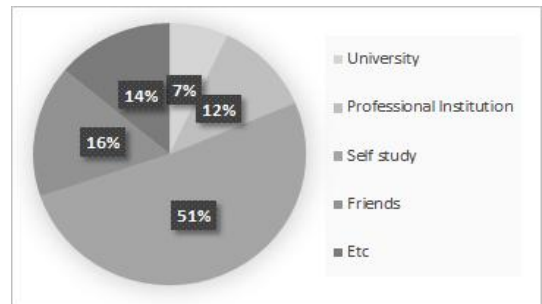


Fig. 6. Learning path of HPC Knowledge & Skill

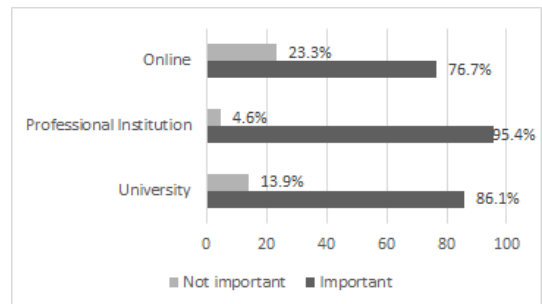


Fig. 7. Importance of HPC education types

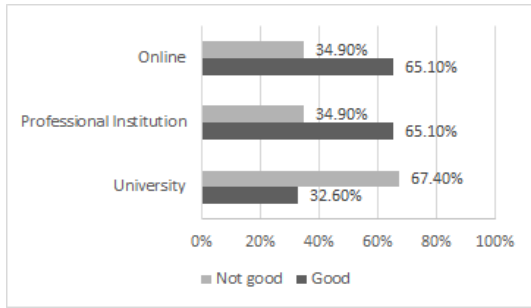


Fig. 8. Degree of provision by HPC educational type

Fig. 8과 같이 각 교육기관 유형별 교육 제공 정도에 대해 ‘대학은 잘 제공한다’는 14명(32.6%), ‘대학은 잘 제공하지 못한다’는 29명(67.4%), ‘전문기관은 잘 제공한다’는 28명(65.1%), ‘전문기관은 잘 제공하지 못한다’는 15명(34.9%), ‘온라인 교육이 잘 제공된다’는 28명(65.1%), ‘온라인 교육이 잘 제공되지 못한다’는 15명(34.9%)로 나타났다.

3.3 교육기관의 중요수준과 제공수준

고성능컴퓨팅 교육을 제공하는 대학, 전문기관, 온라인으로 구분하여 각 제공기관의 중요수준과 현재 제공수준에 대해 설문조사를 실시하여 수집된 자료를 바탕으로 대응표본 t-검증을 실시하였으며 그 결과는 Table 5와 같다. 대학, 전문기관, 온라인 교육제공에 대한 t-검정 결과 모두 현재 제공수준과 요구되는 중요수준 간 차이는 0.55~1.643으로 유의한 차이가 나타났으며 대학의 편차가 제일 크게 나왔다.

Table 5. t-test of HPC education type (n=43)

HPC Education Type	Present Level		Required Importance Level		RL-PL		
	Avg	SD	Avg	SD	Avg	SD	t
University	2.93	1.113	4.57	1.016	1.643	.879	12.129***
Professional institution	3.93	.973	4.98	1.047	1.048	1.209	5.617***
Online	3.85	1.122	4.40	1.236	.550	1.413	2.461*

***p<.001, *p<.05

또한 두 수준간의 단순한 비교를 하는 t-검정의 단점을 보완하기 위해 각 교육제공 유형들의 Borich의 요구도 분석[36]을 실시하였다. Borich 요구도는 필요수준(또

는 중요수준)과 현재수준의 차이를 계산하고 두 수준의 차이에 필요수준의 값을 반영하여 우선순위를 정하는 요구분석 방법으로 제안되었다. 최근에는 Borich의 공식을 바꾸어서 두 수준의 차이를 전체 합산한 값과 요구되는 평균값을 곱한 다음 전체 사계수를 나눈 값을 사용하여 우선순위를 제시하고 있다[37,38].

$$\text{Borich's score} = \frac{\{\sum(RL - PL)\} \times \overline{RL}}{N}$$

RL(Required Level)은 필요한 수준(또는 중요수준)이며, PL(Present Level)은 현재 제공하고 있는 수준이다. \overline{RL} 은 필요수준(또는 중요수준)의 평균값이며 N은 전체 대상수를 의미한다. 따라서 Borich의 우선순위 계산에 의하면 필요수준이 높게 나오고 현재 제공수준이 낮게 나오면 요구도의 값은 더욱 높아지게 된다.

Borich의 요구도 계산식을 사용하여 도출된 요구도 값과 우선순위는 Table 6과 같다. 대학은 7.51로 가장 높고 전문기관은 5.21, 온라인은 2.43을 나타냈다.

Table 6. Borich's score of HPC education type

(n=43)

HPC Education Type	Borich's score	Score Ranking
University	7.51	1
Professional institution	5.21	2
Online	2.43	3

3.4 고성능컴퓨팅 교육 인식조사 결과 분석

인식 조사에서 고성능컴퓨팅 교육의 필요성에 대해 아주 높게 나왔으며, 대학에서 제공하는 교육의 정도에서 67.4%가 부정적으로 조사되었으며 계산과학분야의 전공자들은 51%가 독학으로 고성능컴퓨팅 활용에 대해 지식과 기술을 습득하였다고 조사되었다. 또한 교육기관에 대한 중요수준과 제공수준의 편차와 교육요구도 조사에서 대학이 제일 높게 조사되었다. 이러한 결과는 다음 3가지의 요인으로 분석되었다. 첫째, 계산과학분야의 전공 과목 수 대비 컴퓨터과학 관련 과목의 비율이 너무 낮다는 것이다. 물리 4.3%, 화학 2%, 생명과학 1.4%, 지구과학 3.7%로 아주 낮게 조사되었다. 둘째, 교육과정에 포함된 1개 또는 2개의 IT과목이 기초 프로그래밍 위주의 과목과 컴퓨터를 이용하여 응용분야의 소프트웨어를 활

용하는 과목만으로 구성되어 있는 것이다. 셋째, 외국에 비해 국내에 계산과학과 관련된 학과도 매우 적은 것이다. 물리, 화학 등 순수과학과 순수과학을 응용하는 계산과학과의 분리가 필요하다.

4. 고성능컴퓨팅 교육을 위한 정책적인 제언

앞의 자료 조사 및 설문 조사한 결과를 통해 분석한 의미는 다음과 같다.

첫째, 모든 대학이 아닌 최소한 지역 대학별 고성능컴퓨팅 활용 역량을 확보하기 위한 계산과학 관련 학과가 추가되어야 하며 자연과학대학의 물리학, 화학, 생명과학, 지구과학의 교육과정에 체계적인 IT교육들이 도입되어 한다. 해외에서는 많은 대학들이 고성능컴퓨팅 활용과 관련된 교육과정을 제공하고 있으나 국내는 많이 부족한 상황으로 조사되었다. 따라서 계산과학분야의 고성능컴퓨팅 역량을 확보할 수 있도록 여러 대학에서 IT교육 내용을 개발하여 컴퓨터과학의 과목을 그대로 추가하거나 기초 프로그래밍만 제공하는 것이 아닌 컴퓨터과학의 핵심 주제들을 선별하여 기본적인 개념을 이해하고 다양한 응용분야의 실무에 적용할 수 있는 역량을 확보하는데 중점을 두도록 한다. 예를 들면 컴퓨터과학의 과목이 물리학, 화학, 생명과학, 지구과학 등 응용 연구와 융합되어 활용될 수 있도록 학부 1학년과정부터 4학년과정까지의 단계와 대학원 과정의 단계를 구분하여 차이를 두면서 일관성을 가지는 교육과정으로 개발하는 것이다. 대학의 학부과정에서는 기본단계와 응용단계로 구성하고 대학원과정에서는 연구중심의 전문적인 심화단계의 교육과정으로 개발되어야 한다. 이를 통해 학부를 졸업한 후 본인 스스로 각 응용분야의 문제를 적용할 수 있도록 실무역량을 확보할 수 있게 하고 대학원을 졸업한 후에는 연구 분야별 고성능컴퓨팅을 활용하여 특화된 연구를 수행할 수 있는 전문역량을 확보할 수 있게 하여야 한다.

둘째, 국내 지역별로 전문기관들과 연계하여 고성능컴퓨팅 사용자들을 양성할 수 있는 계획과 실행이 필요하다. 현재는 한국과학기술정보연구원의 국가슈퍼컴퓨팅센터와 기상분야의 기상청 슈퍼컴퓨팅센터에서 자체적으로 교육을 제공하고 있지만 미국이나 유럽처럼 대학과

연계된 교육 훈련은 미비하다. 앞의 조사에서도 전문기관 역할의 중요도가 95.4%로 높게 보여주고 있는 것을 반영하여 대학과 다른 역할로 되어야 한다. 전문기관의 교육과정은 대학에서 진행되는 기초적인 이론과 개념보다는 단계적으로 고성능컴퓨팅 활용을 위한 재교육 및 최신 기술 등 연구 및 산업현장의 요구를 반영하여야 한다. 또한 중·장기적으로 고성능컴퓨팅의 기초부터 응용까지 활용할 수 있는 현장중심의 기초 및 고급의 교육과정이 개발되어야 한다. 그리고 다양한 분야에 특화된 교육과정이 하나의 전문기관에 집중되기보다는 지역의 특성을 반영한 몇 개의 거점센터로 분산되어 지역 대학과 협력하도록 체계를 갖추는 것도 필요하다.

셋째, 양질의 고성능컴퓨팅 교육 내용을 확보하여 오프라인 교육에 참여하지 못하는 학생들과 현업 실무자들을 위한 온라인 교육환경을 구축해야 한다. 앞의 조사에서도 온라인 교육의 중요도가 76.7%로 나타날 정도로 온라인 교육에 대한 요구가 높다. 해외 VSCSE(Virtual School of Computational Science and Engineering)에서는 매년 시행된 교육 자료를 저장한 후 온라인으로 제공[39]한다. 또한 MOOC를 통해 스스로 고성능컴퓨팅을 배우고 실습할 수 있도록 교육과정을 개발하는 연구[40]도 진행되고 있다. 국내에서는 한국과학기술정보연구원에서 일부 온라인으로 교육 코스를 제공[34,41]하고 있지만 많이 제한적이다. 이에 여러 전문기관과 연계하여 각 기관에서 제공한 교육 콘텐츠를 수집하여 단계별, 수준을 고려하여 다양한 사용자들이 독학이 가능한 교육과정을 제공할 필요가 제기된다.

5. 결론

본 연구는 계산과학분야의 IT교육 현황 조사와 고성능컴퓨팅 교육을 개선하기 위한 인식 조사를 수행하였다. 국내 고성능컴퓨팅 활용을 위한 계산과학 관련 학과가 매우 적었으며 물리학, 화학, 생명과학, 지구과학 등 계산과학분야에서도 고성능컴퓨팅 활용을 위한 IT교육은 매우 부족한 실정이었다. 인식조사에서 고성능컴퓨팅 교육의 필요성은 높게 나타났으나 국내의 고성능컴퓨팅 교육 환경은 적절하지 않다고 응답하였다. 또한 교육요구도에 대해 전문기관과 온라인 뿐 만 아니라 대학의 역할이 제일 높게 조사되었다. 이는 고성능컴퓨팅 교육현황과 사

용자들의 인식이 일치하다는 것을 보여주고 있다.

끝으로 본 연구를 통해 두 가지 시사점을 제공한다. 첫째 대학을 통해 고성능컴퓨팅을 활용할 수 있는 역량을 확보하기 위한 교육이 제일 시급한 것으로 조사되었다. 대학의 교육 제공 환경이 미흡하지만 공통적인 교육모델과 핵심 주제들은 선정하여 기초단계·응용단계 등에 따른 교육과정을 개발하고 계산과학분야뿐만 아니라 다른 학문분야에서도 융합될 수 있도록 해야 한다. 둘째 전문 기관과 온라인을 통한 교육제공은 대체로 긍정적으로 조사되었지만 여전히 역할의 중요도에서 높게 나온 점은 새로운 니즈를 반영해야 할 필요가 있다. 즉 고성능컴퓨팅 관련 기술이 점점 발전하고 새로운 소프트웨어 기술들이 나오고 있지만 사용자들이 대학의 교육과정에서 이러한 지식과 기술들을 배우기가 쉽지 않다. 따라서 최신 기술을 반영하고 특화된 산업 수요에 따라 단기·중기 과정의 실무 교육 프로그램이 전문기관을 통해 집중적으로 개발·제공되어야 하며 양질의 고성능컴퓨팅 교육 콘텐츠를 지속적으로 확보하여 온라인 교육을 통해 기본적인 내용에 대해 독학이 가능한 교육 환경을 구축하고 교육 서비스가 제공되어야 한다.

하지만 본 연구에서는 일부 한계점을 가지고 있다. 계산과학분야의 고성능컴퓨팅 교육 방향이 주요 내용이기 때문에 산업, 사회, 공학 등 다른 분야에 융·복합되는 연구로 확장되어야 한다. 또한 교육방향을 활용중심으로 제한하여 고성능컴퓨팅을 개발하고 연구하기 위한 교육은 제외되었다. 따라서 활용뿐만 아니라 연구·개발에서 요구하는 교육에 관한 연구도 필요하다.

본 연구를 기초로 하여 대학 중심의 고성능컴퓨팅 활용 역량을 확보하기 위한 구체적인 교육 과정 개발에 관한 후속 연구가 필요하다. 학부과정에서 제공되어야 할 지식과 기술들은 무엇인지와 대학원과정에서 제공되어야 할 지식과 기술들은 무엇인지에 대한 연구가 지속되어야 한다. 이에 통해 계산과학분야뿐만 아니라 여러 분야에서 고성능컴퓨팅 역량을 확보할 수 있는 교육체계를 설계하고 개발할 수 있을 것으로 기대된다.

REFERENCES

- [1] D. E. Stevenson. (1993, March). Science, computational science, and computer science: at a crossroads. In *Proceedings of the 1993 ACM conference on Computer science* (pp. 7-14). ACM.
DOI : 10.1145/170791.170795
- [2] E. I. D. C. Joseph. (2015). High Performance Computing in the EU: Progress on the Implementation of the European HPC Strategy Report. *PATC OMB, PRACE Advanced Training Centres HPC Curriculum 2013-2014, PRACE IP3 April 2014, 2013*.
- [3] J. M. Shalf & R. Leland. (2015). Computing beyond moore's law. *Computer*, 48(12), 14-23.
DOI : 10.1109/MC.2015.374
- [4] S. J. Ezell & R. D. Atkinson. (2016). The vital importance of high-performance computing to US competitiveness. *Information Technology and Innovation Foundation, April, 28*.
- [5] O. Yasar & R. H. Landau. (2003). Elements of computational science and engineering education. *SIAM review*, 45(4), 787-805.
DOI : 10.1137/S0036144502408075
- [6] O. Yasar, K. S. Rajasethupathy, R. E. Tuzun, R. A. McCoy & J. Harkin. (2000). A new perspective on computational science education. *Computing in Science & Engineering*, 2(5), 74-79.
DOI : 10.1109/MCSE.2000.10020
- [7] C. Taddonki. (2012, May). Basic parallel and distributed computing curriculum, Second NSF. In *TCPP Workshop on Parallel and Distributed Computing Education (EduPar'12), in conjunction with the 26th IEEE International Parallel & Distributed Processing Symposium (IPDPS)*.
- [8] L. A. Wilson & S. C. Dey. (2016, November). Computational science education focused on future domain scientists. In *Proceedings of the Workshop on Education for High Performance Computing* (pp. 19-24). IEEE Press.
DOI : 10.1109/EduHPC.2016.8
- [9] C. Connor, A. Bonnie, G. Grider & A. Jacobson. (2016, November). Next generation HPC workforce development: the computer system, cluster, and networking summer institute. In *Proceedings of the Workshop on Education for High Performance Computing* (pp. 32-39). IEEE Press.
DOI : 10.1109/EduHPC.2016.10
- [10] National Assembly, *National Ultra High Performance Computers Utilization and Promotion Law*, <http://likms.assembly.go.kr/law/lawsLawInqyDet11010.do>
- [11] T. Ungerer & P. Carpenter. (2018). Eurolab-4-HPC Long-Term Vision on High-Performance Computing. *arXiv preprint arXiv:1807.04521*.

[1] D. E. Stevenson. (1993, March). Science, computational science, and computer science: at a crossroads. In *Proceedings of the 1993 ACM conference on Computer*

- [12] Y. J. Kim. (February 26, 2012) *Ultra high-performance computing to lead the fourth industrial revolution*, Etnews, <http://www.etnews.com/20180223000182>
- [13] S. Gallopoulos, E. N. Houstis & J. R. Rice. (1992). Future research directions in problem solving environments for computational science.
- [14] U. Fabricius, C. Freundl, H. Köstler & U. Rude. (2005, May). High performance computing education for students in computational engineering. In *International Conference on Computational Science* (pp. 27-35). Springer, Berlin, Heidelberg.
DOI : 10.1007/11428848_4
- [15] S. W. G. O. C. U. EDUCATION, P. T. A. L. P. Co-Chairs, A. Shiflet, I. Vakalis, K. Jordan & S. S. John. (2011). Undergraduate computational science and engineering education. *SIAM review*, 53(3), 561-574.
DOI : 10.1137/07070406X
- [16] V. Gergel, I. Meyerov & A. Sysoyev. (2015). Unified Assessment of Skills in Parallel and Distributed Computing. *Fac. Comput. Math. Cybern*, 5-6.
- [17] EUROLAB-4-HPC. *D3.2 Best Practices in HPC Training*. <https://www.eurolab4hpc.eu/>
- [18] R. H. Landau. (2007). Computational Physics Education: why, what and how. *Computer physics communications*, 177(1-2), 191-194.
DOI : 10.1016/j.cpc.2007.02.040
- [19] *HPC Students Undergraduate Programs*. <http://hpcuniversity.org/students/undergraduatePrograms/>
- [20] NSF. (2011), *Course Catalog*, XSEDE USER PORTAL, <https://portal.xsede.org/training/course-catalog>
- [21] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gauthier, A. Grimshaw & R. Roskies. (2014). XSEDE: accelerating scientific discovery. *Computing in Science & Engineering*, 16(5), 62-74.
DOI : 10.1109/MCSE.2014.80
- [22] Texas Advanced Computing Center, The University of Texas at Austin, (2011-2018), *ACADEMIC COURSES*, <https://www.tacc.utexas.edu/education/academic-courses>
- [23] S. Lathrop, A. Mason, S. I. Gordon & M. Faerman. (2013, July). HPC university: getting information about computational science professional and educational resources and opportunities for engagement. In *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery* (p. 67). ACM.
DOI : 10.1145/2484762.2484771
- [24] *HPC University*. <http://hpcuniversity.org/about>
- [25] PRACE, *PRACE Members*. <http://www.prace-ri.eu/members/>
- [26] *Training Portal of Partnership for Advanced Computing in Europe(PRACE)*. http://www.training.prace-ri.eu/uploads/tx_pracetmo
- [27] N. Alexandrov, V. Alexandrov & R. Ramirez. (2012). The Role of Computational Science and Emerging Technologies in the Natural Sciences Education at University Level. *Procedia Computer Science*, 9, 1789-1798.
DOI : 10.1016/j.procs.2012.04.197
- [28] EPCC, *Learning with EPCC*. <http://www.epcc.ed.ac.uk/education-training/>
- [29] High-Performance Computing Center(HLRS). *Solutions&Services/TRAINING*, <http://www.hlrs.de/solutions-services/service-portfolio/training/>
- [30] BSC Education, *Education*, <https://www.bsc.es/education>
- [31] S. J. Kim & D. E. Cho. (2018). A Study on Learning Model for Effective Coding Education. *Journal of the Korea Convergence Society*, 9(2), 7-12.
- [32] K. M. Kim & H. S. Kim. (2014). A Case Study on Necessity of Computer Programming for Interdisciplinary Education. *Journal of digital convergence*, 12(11), 339-348.
- [33] Supercomputing education center, *About Supercomputing courses* <https://webedu.ksc.re.kr/subjectIntroduce.es?mid=a10201000000>
- [34] EDISION, *EDISION Portal* <https://www.edison.re.kr/web/portal/>
- [35] Ministry of Education, *Standard classification system of Higher Education in Korea* <http://www.academyinfo.go.kr/UIPISA/uipnh/unt/mjrinfo/UntMjrInfoByUnvList.do>
- [36] G. D. Borich. (1980). A needs assessment model for conducting follow-up studies. *Journal of teacher education*, 31(3), 39-42.
DOI : 10.1177/002248718003100310
- [37] S. K. Oh, J. S. Jun & Y. H. Parl. (2014). Complementing a Typical Educational Needs Analysis Using a Survey in Setting the Priority of the Needs. *Journal of elementary education research*, 27(4), 77-97.
- [38] M. R. Eom, W. J. Choi & Y. H. Song. (2018). Exploring the Educational Needs of Learning Supporting Program on the Students' Perception of Current Competencies and Important Competencies. *Journal of Convergence for Information Technology*, 8(3), 175-181.

DOI :10.22156/CS4SMB.2018.8.3.175

- [39] K. J. Cahill, S. Lathrop & S. Gordon. (2017). Building a Community of Practice to Prepare the HPC Workforce. *Procedia computer science*, 108, 2131-2140.
DOI : 10.1016/j.procs.2017.05.059
- [40] J. Mullen, C. Byun, V. Gadepally, S. Samsi, A. Reuther & J. Kepner. (2017). Learning by doing, High Performance Computing education in the MOOC era. *Journal of Parallel and Distributed Computing*, 105, 105-115.
DOI : 10.1016/j.jpdc.2017.01.015
- [41] Supercomputing education center, *online course*.
<https://webedu.ksc.re.kr/movieLecture.es?mid=a10401000000>

윤 희 준(Yoon, Hee Jun)

[정회원]



- 1997년 2월 : 충남대학교 컴퓨터공학과(석사)
- 2012년 3월 ~ 현재 : 성균관대학교 대학원 교과교육과 박사과정
- 2000년 6월 ~ 현재 : 한국과학기술정보연구원

- 관심분야 : 컴퓨터 교육, 고성능컴퓨팅, SW교육
- E-Mail : k2@kisti.re.kr

안 성 진(Ahn, Seong Jin)

[정회원]



- 1988년 2월 : 성균관대학교 1988년 정보공학과(학사)
- 1990년 2월 : 성균관대학교 정보공학과(석사)
- 1998년 2월 : 성균관대학교 정보공학과(박사)

- 1999년 2월 ~ 현재 : 성균관대학교 컴퓨터교육과 교수
- 관심분야 : 컴퓨터 교육, SW교육, 정보윤리, 정보보안
- E-Mail : sjahn@skku.edu