

# 교육요구분석을 통한 계산과학분야의 고성능컴퓨팅 지식요소에 관한 연구

윤희준\* · 안성진\*\*

성균관대학교 교과교육학과\* · 성균관대학교 컴퓨터교육과\*\*

## 요 약

본 연구는 계산과학분야의 고성능컴퓨팅 교육을 위한 지식요소들을 제안하고자 한다. 이를 위해 고성능컴퓨팅 전문가들을 대상으로 1차 설문을 통해 내용타당도와 신뢰도를 조사하여 20개의 후보지식요소들을 도출하였다. 고성능컴퓨팅 사용자들을 대상으로 2차 설문을 통해 후보지식요소들에 대해 t-test, Borich 요구도, The Locus for Focus 모델을 적용하여 고성능컴퓨팅 교육을 위한 10개의 지식요소들을 도출하였다. 그 결과 고성능컴퓨팅 교육을 위한 '기본 병렬성', '병렬성', '병렬통신 및 조정', '병렬분할', '병렬 알고리즘 분석 및 프로그래밍', '모델링 및 시뮬레이션 소개' 6개의 1순위 지식요소들과 '기본 프로그래밍 개념', '기본 자료구조', '메모리 관리', '알고리즘 및 설계' 4개의 2순위 지식요소들을 도출하였다.

키워드 : 고성능컴퓨팅, 고성능컴퓨팅 교육, 지식요소, 교육요구도, 교육과정

## A Study on the Knowledge Elements of HPC in Computational Science through Analysis of Educational Needs

Heejun Yoon\* · Seongjin Ahn\*\*

Dept. of Disciplinary Education, Sungkyunkwan University\* ·

Dept. of Computer Education, Sungkyunkwan University\*\*

## ABSTRACT

The purpose of this study is to suggest the knowledge elements for HPC education in computational science. For this purpose, the survey for HPC experts was conducted to verify the content validity and reliability, and the 20 candidate knowledge elements was extracted. And the second survey for HPC users was conducted to apply the t test, Borich requirement, and The Locus for Focus model. And 10 knowledge elements for HPC education were derived. As a result, the first group was 'Parallelism Fundamentals', 'Parallelism', 'Parallel communication and coordination', 'Parallel Decomposition', 'Parallel Algorithms, Analysis, and Programming' and 'Introduction to Modeling and Simulation', 'Fundamental Programming Concepts', 'Fundamental Data Structures', 'Memory Management', 'Algorithms and Design' were second group for HPC education.

Keywords : HPC, HPC Education, Knowledge Elements, Educational Needs, Curriculum

교신저자 : 안성진(성균관대학교 컴퓨터교육과)

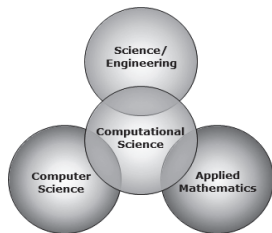
논문투고 : 2018-10-16

논문심사 : 2018-10-23

심사완료 : 2018-10-25

## 1. 서론

초기의 계산과학(Computational Science)은 물리, 천문, 지구과학 분야의 거대한 문제를 해결하기 위한 도전이었으며[1] 과학적인 난제뿐만 아니라 공학, 산업 등에서 제기되는 실용적인 응용문제를 해결하기 위해 과학, 공학, 수학, 그리고 컴퓨터과학이 (Fig. 1)처럼 서로 융합된 학문으로 빠르게 발전하였다[2]. 이는 모의실험을 통해 복잡한 문제에 대해 빨리 계산하여 원하는 결과를 얻을 수 있는 고성능컴퓨팅이 계산과학의 중요한 요소라는 것을 의미한다[3]. 이러한 고성능컴퓨팅의 중요한 역할은 고성능컴퓨팅이 발전할수록[4] 거대한 문제를 계산하는 과학 영역에서 효율성과 생산성을 높이는 산업 영역, 대용량 데이터의 생산·분석·활용되는 데이터 기반의 4차 산업 영역 등으로 확대되고 있다[5].



(Fig. 1) Relationship of CSE to other disciplines.

미국의 고성능컴퓨팅법[6]에 따르면 워크스테이션, 슈퍼컴퓨터 시스템(백터 슈퍼컴퓨터와 대규모 병렬 시스템 포함), 고용량 고속망, 특수 목적 및 실험적 시스템, 응용 및 시스템 소프트웨어를 포함하는 첨단 컴퓨팅, 통신 및 정보 기술로 정의하고 있으며, 국내에서는 국가초고성능컴퓨터 활용 및 육성에 관한 법률[7]에 초고성능컴퓨터나 초고성능컴퓨터 기술을 이용한 고용량·고속의 전산망의 활용, 특수 목적의 실험시스템의 구축, 응용 및 시스템 소프트웨어, 대용량 데이터 관리 등을 포함하는 컴퓨팅, 통신 및 정보기술로 정의하고 있다. 최신 IT기술의 집합체인 고성능컴퓨팅은 특정한 구조의 CPU, Storage, Network 등 하드웨어와 프로그래밍 언어, 컴파일러, 라이브러리, 디버거, 테스트 코드 등 소프트웨어 및 개발 환경들로 구성되어 있다[8]. 따라서 계산과학자들은 컴퓨터과학의 기본적인 지식과 기술들뿐

만 아니라 복잡한 문제를 해결하기 위해 고성능컴퓨팅과 관련된 광범위한 지식과 기술들을 배워야 한다[9]. 예를 들면 계산과학자들은 본인들이 사용해야 할 고성능컴퓨터의 하드웨어의 구조와 'Computational science tools', 'High-performance computing', 'Computer applications'와 같은 여러 지식영역(Knowledge Areas)에 속하는 과목들의 지식과 기술들을 배워야 한다[10]. 또한 컴퓨터과학의 모든 과목들이 중요하지만 계산과학 전공자들이 컴퓨터과학의 많은 과목들을 배우는 것이 현실적으로 한계가 있기에[11] 꼭 배워야 할 과목들을 선별하고 순서를 정하는 체계화도 필요하다.

따라서 본 논문은 교육요구분석을 통해 계산과학분야의 고성능컴퓨팅 교육을 위한 지식요소들을 제시하려 한다. 컴퓨터과학의 많은 지식요소 중에서 계산과학분야에서 요구하는 지식요소들은 무엇인지를 알아보기 위해 미국의 컴퓨터과학 교육과정(Computer Science Curricula 2013, 이하 CS2013)[12]을 기초자료로 활용하여 전문가 대상의 1차 설문을 통해 후보 지식요소들을 도출한 후 도출된 후보 지식요소들에 대해 고성능컴퓨팅 사용자 대상의 2차 설문을 실시한 후 t-검정, Borich 요구도 분석[13], The Locus for Focus 모델을 적용하여 교육요구분석[14]을 한 결과를 토대로 계산과학분야의 고성능컴퓨팅 교육을 위한 지식요소들을 제시하였다.

## 2. 고성능컴퓨팅 교육 관련 연구

문헌조사를 통해 1990년대부터 최근까지 진행된 연구들을 분석하여 교육 주제, 자료 수집 방법, 자료 분석 방법, 연구 주체에 대해서 정리한 내용을 <Table 1>에 나타내었다. 선행연구를 통해 4가지를 확인하였다. 첫째, 고성능컴퓨팅 교육의 주제가 컴퓨터과학의 기본적인 내용에서 고성능컴퓨팅의 세부 주제로 다양화되고 있다는 것이다. 2000년대 중반까지 각 연구에서 제시한 교육의 주제는 컴퓨터과학의 프로그래밍, 자료구조, 알고리즘 등 기본적인 주제들을 제시하였으며 2010년 이후부터 컴퓨터과학의 기본적인 주제가 아닌 병렬 프로그래밍, 병렬 컴퓨팅, 병렬성능, 병렬 구조 등 병렬에 관한 세부 주제가 많이 제안되었으며 추가로 디버깅, 최적화, 가시화, I/O Tuning 등의 주제도 제안되었다. 둘째, 교육 주

<Table 1> Researches for HPC Education

No	Year	Educational Topic(Knowledge & Skills)	Data Collection	Analysis Method	WHO
1[15]	1992	Object-oriented tech., Software Interconnection Tech., Language, Compiler, Parallel and Distributed Computation	Problem Solving Example	Case study	University
2[10]	2000	programming, operating system, performance, computer architecture, parallel language	Curriculum	Curriculum Analysis	University
3[16]	2003	Programming, Performance benchmarks, Relationship among architecture/language/performance, parallel libraries	Lecture Survey	Evaluation	University
4[17]	2005	Algorithms, Data Structure, Parallel computing, Parallel Algorithms, Programming, Computer architecture, code profiling, Tuning, optimization	Lecture Survey	Evaluation	University
5[18]	2006	Developing HPC Code, Debugging, Testing and Verification & Validating, Optimizing Code, Scheduling	FGI	Case study	Industry
6[19]	2007	Programming & Algorithms, Architectures (parallel/distributed/grid), Performance analysis, Software engineering, Data Analysis & Post processing	Education Survey	Exert Interview	Virtual Group
7[20]	2011	Programming, Algorithms, High-level Language, Data structures, Parallel programming(MPI, OpenMP),	Summary Article	Case study	SIAM Group
8[21]	2015	computer programming(MPI, OpenMP), Concepts of Concurrency, Finding/Evaluating Parallelism, Decomposition, Parallel I/O	Online Lecture Survey	Feedback	University
9[22]	2016	Parallel computing concepts & architecture(HPC, HTC), Programming models for parallel computers, large-scale systems	Lecture Survey	Evaluation	University
10[23]	2016	Software engineering, Visualization, Programming model, Parallel programming, Performance, Debugging, I/O Tuning	Online Lecture Survey	Evaluation	University
11[24]	2017	Introduction of HPC, Algorithms for Scalable Many-core Computing, Extreme Scale Systems, High Performance Visualization	Online Lecture Survey	Evaluation	Super Computing Center
12[25]	2018	Parallel programming, Algorithms, Computational complexity, High-level languages, Data structures, Scientific visualization, Software engineering tools	Success Example	Case study	SIAM Group

제를 도출하기 위한 자료수집 방법이 대부분 강의 후 강의 참여자를 대상으로 한 강의 평가 설문을 통해 수집되었으며, 일부 연구에서는 문제 해결, 성공사례, 고성능컴퓨팅 교육 관련 논문, 계산과학 교육과정을 정리하여 수집하였다. 하지만, 자료를 수집한 설문 도구에 대한 구체적인 내용은 명시되지 않았다. 셋째, 자료 분석 방법이 강의 평가 자료에 대한 일반적인 통계(ex. 빈도 분석)와 사례연구에 대해 전문가 그룹 또는 워크숍을 통한 정성적인 방법으로 되었다. 넷째, 연구를 진행한 주제들은 계산과학 관련 강의를 제공하는 대학들이 가장 많았으며, 1952년에 설립된 미국산업응용수학회(SIAM)[26]와 같이 고성능컴퓨팅을 활용하는 학회나 고

성능컴퓨팅을 운영하는 컴퓨팅센터 등이 연구를 하였다.

대부분 선행연구들은 강의 수강자들을 대상으로 강의 평가에 대한 설문조사와 사례연구를 통해 자료를 수집한 후 일반적인 통계와 전문가 그룹 중심의 정성적인 평가(인터뷰, 회의 등)를 통해 고성능컴퓨팅 교육을 위한 주제들을 도출하였다. 대부분의 제안된 주제들은 컴퓨터과학의 교육과정과 비슷한 내용을 포함하고 있었으나, 최근에는 고성능컴퓨팅의 세부내용으로 된 지식요소와 일부 디버깅, 최적화, SE Tools 등과 같은 소프트웨어 개발과 사용에 관한 기술요소가 혼용되어 제시된 연구도 있었다.

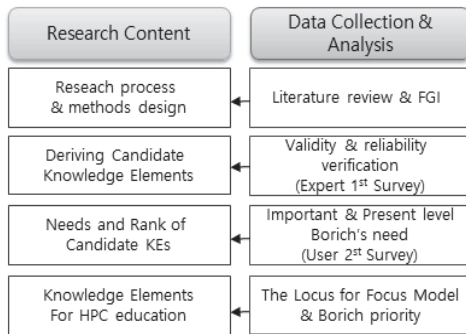
국내에서는 SW교육 콘텐츠 개발[27] 연구, 프로그래

명 교육 연구[28] 등은 활발하나 고성능컴퓨팅 교육에 관한 연구는 매우 저조한 상태이다.

### 3. 연구 방법

본 연구는 3가지 관점에서 선행연구와 구별된다. 첫째, 수집된 자료가 관련분야 전문가 대상으로 1차 설문조사를 수행한 단계는 비슷하지만, 강의 수강자와 달리 현재 고성능컴퓨팅 사용자를 대상으로 2차 설문조사를 실시하여 자료를 수집한 점이다. 둘째, 명확한 지식요소들을 제안하기 위해 미국 컴퓨터과학의 교육과정(CS2013)을 도구로 활용한 점, 셋째, 자료분석 방법이 고성능컴퓨팅 교육에 필요한 지식요소에 대한 중요수준과 본인의 성취수준을 조사하고 우선순위를 정하기 위한 Borich 요구도와 The Locus for Focus 모델을 적용하여 분석[14][31]하여 우선적인 지식요소들을 제안한 점이다.

#### 3.1 연구절차



(Fig. 2) The Research Process

요구분석을 통한 고성능컴퓨팅 교육을 위한 지식요소들을 도출을 위한 수립된 연구 절차는 (Fig. 2)에서 보여주고 있다. 먼저 교육 분야 전문가 2명(교수, 박사), 고성능컴퓨팅의 IT전문가 2명(박사), 고성능컴퓨팅 활용 전문가 3명(박사)으로 구성된 FGI를 통해, 기초자료인 CS2013를 활용한 설문도구의 활용, 1차 설문 및 2차 설문의 연구대상, 방법 등 전체적인 연구 방향, 내용, 절차 등을 구체화하였다. 1차 설문조사는 고성능컴퓨팅 전문

가들을 대상으로 실시하여 후보 지식요소들을 도출한 후, 2차 설문조사를 통해 1차에서 도출된 해당 지식요소들을 가지고 고성능컴퓨팅 사용자들을 대상으로 한 2차 설문조사를 실시하였다. 2차 설문 결과에 대해 Borich 요구도 순위와 The Locus for Focus 모델을 적용하여 1순위 및 2순위 지식요소들을 도출하였다.

#### 3.2 연구대상

<Table 2> Characteristics of Second Survey

n=80

Characteristic	Category	Count(%)
Gender	Male	66(82.5)
	Female	14(17.5)
Age	20s	19(23.7)
	30s	32(40)
	40s	25(31.3)
	50s	4(5)
Education level	Doctor's degree	45(56.3)
	Master's degree	15(18.7)
	Bachelor; degree	20(25)
Major	Physics	34(42.5)
	Chemistry.	8(10)
	Biological Sciences	8(10)
	Earth/Atmospheric science	9(11.3)
	Etc(IT, Engineering)	12(15.0)
	No answer	9(11.2)
Organization	University	41(51.2)
	Research/Public institute	31(38.8)
	Industry	8(10.0)
Field of application	Physics	31(38.8)
	Chemistry	11(13.8)
	Biological Sciences	8(10.0)
	Earth/Atmospheric science	13(16.3)
	Material Engineering	7(8.8)
	Mechanical Engineering	6(7.5)
Experience using HPC	Etc	4(5)
	~ 3year	26(32.5)
	~ 6year	14(17.5)
	~ 9year	15(18.8)
	9year ~	25(31.2)

1차 설문조사는 2017년 10월 3주 동안 실시하였다. 설문 조사대상은 박사25명(83%), 석사5명(17%)으로 구성된 고성능컴퓨팅 전문가들이며, 대상자들의 전공은 IT전공 20명(66.7%), 응용분야전공 10명(33.3%)이고, 소

속은 대학 7명(23%), 전문기관 23명(77%)으로 구성되었다. 직접 면담 및 이메일을 통해 2017년 10월 3주 동안 설문조사를 실시하였다.

2차 설문조사는 2018년 6월말부터 7월 말까지 고성능 컴퓨팅을 활용하고 있는 사용자 90명을 대상으로 면담 조사, 이메일을 통해 실시하였고 86부(약 95%)가 회수되었다. 일부 자료가 불성실하게 응답한 경우와 대상자가 학부생인 경우인 6부를 제외하고 80부의 자료를 분석하였다. 2차 설문자들의 특성은 <Table 2>에 나타내었다.

### 3.3 CS2013을 활용한 설문도구

계산과학은 컴퓨터과학이 포함된 다분야 학문[2]이기에 컴퓨터과학을 공부하는 대학생들을 위한 커리큘럼으로 새로 개편된 미국의 CS2013 교육과정을 설문 구성을 위한 기초자료로 활용하였다. CS2013은 컴퓨터과학 전공자들이 배워야 할 지식단위(Knowledge Unit, 이하 KU)들이 여러 지식영역들에 포함될 수 있도록 구성된 여러 지식영역(Knowledge Areas, 이하 KA)의 집합체이다. CS2013에는 고성능컴퓨팅과 관련된 'Parallel and Distributed Computing'을 포함한 19개의 지식영역이 제시되어 있으며, 이 중에서 핵심이수 시간을 가진 89개 지식단위들을 조사항목으로 사용하였다.

1차 설문도구에는 89개의 지식단위(KU)들에 대해 간략한 설명, 고성능컴퓨팅 교육을 위한 지식단위들의 중요수준으로 구성하였으며 5점 리커트 척도(매우 중요하지 않다:1, 매우 중요하다:5)로 높은 점수 일수록 고성능컴퓨팅 교육에서 중요한 요소가 되도록 설계하였다. 2차 설문도구는 1차에서 도출된 후보 지식요소들에 대한 구체적인 설명, 중요수준, 본인의 성취수준으로 구성하였으며 6점 리커트(Likert)척도(매우 낮다:1점, 매우 높다:6 점)로 높은 점수 일수록 고성능컴퓨팅 교육에서 중요한 요소이며, 본인이 성취한 수준이 높은 것으로 나타내도록 설계하였다.

### 3.4 자료분석 방법

1차 자료 분석은 고성능컴퓨팅 교육을 위한 지식요소들의 타당도를 검증하기 위해 수집된 1차 설문결과와 내용타당도 지수(Context Validity Ratio: CVR, 이하

CVR)를 산출하였다. 본 연구는 Lawshe(1975)이 제시한 모델을 사용하여 내용타당도 비율(CVR)을 산출하였으며[29], 내용타당도를 계산하는 공식은 아래와 같다.

$$CVR = (N_e - N/2) / (N/2)$$

여기서 N은 조사에 참여하여 응답한 전체 전문가의 수로서 30이며 Ne는 문항이 타당하다고 응답한 수를 말한다. CVR이 0보다 작을 경우는 50% 이하가 타당하다고 응답한 것이고 CVR이 0일 경우는 50% 타당하다고 응답한 것이며 0보다 클 경우는 50% 이상이 타당하다고 응답한 것이다. CVR를 통한 타당도의 기준값은 참여한 전문가의 수에 따라 조정된다[29][30].

또한 신뢰도를 측정하기 위해 각 지식영역에 포함된 지식단위들을 그룹화 하여 Cronbach's  $\alpha$ 값을 측정하였다. 지식단위가 1개인 Computational Science 지식영역과 Graphics and Visualization 지식영역은 신뢰도를 측정하지 못하였다. Cronbach's  $\alpha$ 값은 1에 가까울수록 높은 신뢰도를 가졌다고 판단하고 .7이상일 때 '어느 정도 신뢰성이 있다'라고 판단하는 게 일반적이다.

2차 자료 분석은 고성능컴퓨팅 교육에 대한 요구도를 분석하기 위해 조대연의 연구[14]를 활용하여 현재수준과 중요수준을 Borich 요구도, The Locus for Focus 모델을 적용하여 분석하였다. 세부적인 분석방법은 다음과 같다.

첫째, 대응표본 t-검증을 통해 고성능컴퓨팅 지식요소에 대한 현재수준과 중요수준간의 차이가 있는지 유의미하게 통계적으로 확인한다.

둘째, Borich의 요구도 분석을 통해 후보지식요소에 대한 Borich 요구도 지수를 산출한 후 우선순위를 결정한다. Borich 요구도는 중요수준(또는 필요수준)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[31].

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

RL(Require Level)은 요구되는 중요수준이며, PL(Present Level)은 현재 성취한 수준이다.  $\overline{RL}$ 은 중요수준의 평균값이며 N은 전체 대상수를 의미한다. 따라서 Borich의 우선순위는 계산에 따라 중요수준이 높게 나오고 현재 성취수준이 낮게 나오면 요구도의 값은 더욱 높아지게 된다.

셋째, 우선적으로 필요로 하는 지식요소들과 몇 순위까지를 우선적으로 요구되는지를 결정하기 위해, 좌표평면을 이용하여 우선순위를 결정하는 The Locus for Focus 모델(이하 LF분포모델)을 (Fig. 3)처럼 사용하였다. LF분포모델의 특징은 2개의 축으로 구성되어 좌표평면에 문항들의 점수를 계산하여 우선순위를 시각적으로 나타낸다. 중요수준의 평균값인 가로축과 중요한 수준과 현재 성취한 수준의 차이값의 평균값을 세로축에 표시한다.

LH	Quadrant 2 Low Importance & High Discrepancy	Quadrant 1 High Importance & High Discrepancy	HH
LL	Quadrant 3 Low Importance & Low Discrepancy	Quadrant 4 High Importance & Low Discrepancy	HL

(Fig. 3) The Locus for Focus Model

넷째, The locus for focus모델의 LL분면과 LH분면에 포함된 문항들은 제거하고 HH분면과 HL분면에 속한 문항들에 대해 Borich 요구도 분석의 상위 순위에 따라 최우선 순위와 차순위로 결정한다. The locus for Focus모델의 HH분면에 포함된 문항들의 갯수만큼 Borich 요구도의 우선순위가 높은 문항들을 공통으로 최우선 순위로 결정하고 공통되지 않고 HH사분면과 HL사분면에 포함된 문항들을 차순위로 정하였다[14]. HH분면에 있는 것은 중요하면서 성취도가 낮기 때문에

시급하게 요구되는 영역으로 판단할 수 있다.

#### 4. 교육 요구 분석을 통한 고성능컴퓨팅 교육을 위한 지식요소

##### 4.1 후보 지식요소들 도출

지식단위들이 속하는 지식영역에 대해 신뢰도인 Cronbach's  $\alpha$ 값을 측정할 결과를 <Table 3>에 나타내었다. 지식영역 중에서 IAS(Information Assurance and Security)지식영역이 .965로 신뢰도가 가장 높게 나타났고 AL(Algorithms and Complexity)지식영역이 .664로 가장 낮게 나타났으며 다른 지식영역들은 모두 .8이상으로 높은 신뢰도를 보여주고 있다.

<Table 3> Cronbach's  $\alpha$  of Knowledge Areas

KA	Cronbach's $\alpha$	KA	Cronbach's $\alpha$	KA	Cronbach's $\alpha$
AL	.664	IAS	.965	PD	.835
AR	.835	IM	.834	PL	.865
CN	-	IS	.934	SDF	.867
DS	.929	NC	.921	SE	.938
GV	-	OS	.930	SF	.908
HCI	.907	PBD	-	SP	.899

지식단위들에 대해 내용타당도(CVR)를 조사한 결과는 <Table 4>와 같다. Lawshe(1975)는 응답한 전체 전문가의 수에 따라 최소로 요구하는 CVR값을 제시하였는데 본 연구에서는 전문가의 수가 30인이므로 CVR 값이 .33보다 높아야 타당성이 있다[30]. 또한 CVR이 .33

<Table 4> CVR of CS2013 Knowledge Units

KU	CVR	KU	CVR	KU	CVR	KU	CVR	KU	CVR	KU	CVR	KU	CVR	KU	CVR	KU	CVR	KU	CVR
AL1	0.60	CN1	0.87	HCI2	-0.73	IM3	0.13	NC5	-0.33	PD1	1.00	PL1	-0.07	SDF4	-0.07	SE9	-0.13	SF9	-0.07
AL2	0.60	DS1	-0.33	IAS1	-0.20	IS1	0.13	NC6	-0.13	PD2	1.00	PL2	0.07	SE1	-0.40	SF1	0.07	SP1	-0.80
AL3	0.67	DS2	-0.20	IAS2	-0.40	IS2	0.00	NC7	-0.60	PD3	0.67	PL3	-0.27	SE2	-0.27	SF2	-0.47	SP2	-0.60
AL4	-0.07	DS3	-0.60	IAS3	-0.53	IS3	0.00	OS1	0.53	PD4	1.00	PL4	-0.53	SE3	-0.47	SF3	-0.67	SP3	-0.60
AR1	-0.20	DS4	-0.47	IAS4	-0.47	IS4	0.40	OS2	0.47	PD5	0.87	PL5	-0.53	SE4	-0.60	SF4	0.67	SP4	-0.47
AR2	-0.47	DS5	0.13	IAS5	-0.13	NC1	-0.13	OS3	0.40	PD6	0.80	PL6	-0.53	SE5	-0.13	SF5	0.20	SP5	-0.80
AR3	-0.60	DS6	-0.20	IAS6	-0.33	NC2	-0.07	OS4	0.47	PD7	0.80	SDF1	0.67	SE6	-0.53	SF6	0.47	SP6	-0.33
AR4	0.53	GV1	-0.13	IM1	-0.33	NC3	-0.33	OS5	0.67	PD8	0.13	SDF2	0.53	SE7	-0.27	SF7	-0.33	SP7	-0.33
AR5	0.20	HCI1	-0.60	IM2	-0.20	NC4	-0.27	OS6	-0.07	PD9	-0.13	SDF3	0.53	SE8	-0.60	SF8	-0.27		

<Table 5> Cronbach's  $\alpha$  & CVR of Candidate Knowledge Elements for HPC Education

KA	Cronbach's $\alpha$	Knowledge Units(KU)	CVR	KA	Cronbach's $\alpha$	Knowledge Units(KU)	CVR
Computational Science	-	Introduction to Modeling and Simulation	0.87			Parallelism Fundamentals	1
Architecture and Organization	.835	Memory System Organization and Architecture	0.53			Parallel Decomposition	1
Intelligent Systems	.934	Basic Machine Learning	0.4	Parallel and Distributed Computing	.835	Communication and Coordination	0.67
		Overview of Operating Systems	0.53			Parallel Algorithms, Analysis, and Programming	1
Operating Systems	.930	Operating System Principles	0.47			Parallel Architecture	0.87
		Concurrency	0.4			Parallel Performance	0.8
		Scheduling and Dispatch	0.47			Distributed Systems	0.8
		Memory Management	0.67			Algorithms and Design	0.67
Systems Fundamentals	.908	Parallelism	0.67	Software Development Fundamentals	.867	Fundamental Data Structures	0.53
		Resource Allocation and Scheduling	0.47			Fundamental Programming Concepts	0.53

보다 높은 지식단위는 23개가 도출되었지만 신뢰도가 .7 미만인 지식영역 'Algorithms and Complexity'에 속한 3개의 지식단위들은 제외되었다. 따라서 고성능컴퓨팅 교육 요구분석을 위한 후보 지식요소들이 20개가 도출되었으며 그 결과는 <Table 5>과 같다.

4.2 교육요구분석 및 The Locus for Focus 모델 적용

4.2.1 후보지식요소의 중요수준 및 성취수준

20개의 후보지식요소에 대한 중요 수준과 현재 성취수준을 분석한 결과 20개의 후보 지식요소 중에서 제일 중

<Table 6> t-test fo Knowledge Elements for HPC Education

Knowledge Elements	Important level(RL)		Present level(PL)		RL-PL		t-value
	Avg	SD	Avg	SD	Avg	SD	
Fundamental Programming Concepts	5.28	1.02	4.55	1.22	.73	1.17	5.547***
Fundamental Data Structures	4.76	1.31	4.06	1.31	.70	1.35	4.625**
Overview of Operating Systems	3.81	1.22	3.39	1.27	.42	1.32	2.820**
Operating System Principles	3.73	1.14	3.10	1.38	.63	1.33	4.218***
Concurrency	4.06	1.19	3.20	1.37	.86	1.43	5.351***
Scheduling and Dispatch	4.31	1.19	3.24	1.36	1.08	1.47	6.525***
Memory Management	4.61	1.10	3.48	1.37	1.14	1.36	7.497***
Resource Allocation and Scheduling	4.27	1.11	3.38	1.40	.89	1.44	5.465***
Basic Machine Learning	4.30	1.26	3.26	1.57	1.04	1.50	6.202***
Algorithms and Design	4.94	1.05	3.86	1.30	1.08	1.38	6.953***
Parallelism Fundamentals	4.80	1.26	3.76	1.47	1.04	1.53	6.066***
Parallelism	4.86	1.11	3.55	1.54	1.31	1.51	7.795***
Parallel Communication and Coordination	4.50	1.25	3.28	1.59	1.23	1.57	6.993***
Parallel Decomposition	4.54	1.30	3.29	1.47	1.25	1.55	7.228***
Memory System Organization and Architecture	4.34	1.27	3.28	1.58	1.06	1.53	6.215***
Parallel Architecture	4.35	1.23	3.10	1.63	1.25	1.50	7.469***
Parallel Algorithms, Analysis, and Programming	4.61	1.27	3.35	1.62	1.26	1.62	6.968***
Introduction to Modeling and Simulation	4.96	0.98	3.82	1.39	1.14	1.54	6.567***
Distributed Systems	4.16	1.22	3.14	1.51	1.03	1.30	7.041***
Parallel Performance	4.26	1.26	3.20	1.50	1.06	1.32	7.172***

요하게 요구되는 것은 5.28인 ‘기본 프로그래밍 개념’이며 그 다음으로 ‘모델링 및 시뮬레이션 소개’, ‘알고리즘 설계’으로 나타났다. 성취한 수준에 대해서 4.55인 ‘기본 프로그래밍 개념’이 가장 높게 나타났으며 ‘기본 자료구조’, ‘알고리즘 및 설계’으로 나타났다. 그리고 후보지식요소에 대한 사용자들의 중요수준과 현재 성취한 수준을 비교한 결과 <Table 6>과 같이 모두 유의미한 차이를 보였고 중요 수준(RL)와 성취수준(PL)의 평균에 대한 대응표본 t-검정을 조사하였다. 그 결과 t값이 2.82 ~ 7.795을 나타냈으며 ‘운영체제 개요’ 지식요소만  $p < .01$  유의수준에서 그 차이가 유의미하며, 나머지 지식요소들은  $p < .001$  유의수준에서 그 차이가 통계적으로 유의미하게 나타났다. 컴퓨터과학의 기본적인 지식요소들에 대해 중요수준과 성취수준의 차이가 1이하인 반면 병렬성과 관련된 지식요소들의 수준차이가 1이상으로 높게 나타났다.

**4.2.2 후보지식요소의 Borich 요구도 분석**

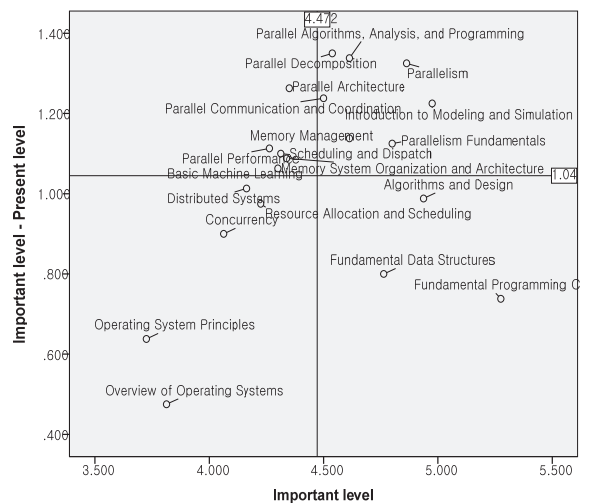
T-검정 후 후보지식요소 20개에 대한 Borich 요구도 공식을 활용하여 분석을 실시하였다. 도출된 우선순위를 분석한 결과는 <Table 7>에 나타내었다. 교육요구도가 높은 지식단위는 ‘병렬성’(6.44)이 가장 높게 나타났으며 다음으로 ‘병렬 알고리즘, 분석 및 프로그래밍’(6.17), ‘병렬 분할’(6.13), ‘모델링 및 시뮬레이션 소개’(6.09) 순으로 높게 나타내었다. 병렬성과 관련된 지식요소들이 순위가 높게 나왔다.

**4.2.3 후보지식요소의 The Locus for Focus 모델 적용 및 분석**

(Fig. 4)에 고성능컴퓨팅 교육에서 중요하게 요구되는 20개의 지식요소에 대해 Locus for Focus Model을 적용한 것을 보여주고 있다. 중요수준과 성취수준 차이와 중요수준의 평균값을 적용하여 HH분면에 속하는 ‘메모리 관리’, ‘병렬성 기본’, ‘병렬성’, ‘병렬통신 및 조정’, ‘병렬분할’, ‘병렬 알고리즘, 분석 및 프로그래밍’, ‘모델링 및 시뮬레이션 소개’로 7개의 지식단위가 도출되었으며 중요도가 높으면서 성취도가 높은 HL분면에 속하는 지식요소는 ‘기본 프로그래밍 개념’, ‘기본 자료 구조’, ‘알고리즘 및 설계’ 3개로 도출되었다.

<Table 7> Borich needs & Rank

No	Knowledge Elements	Borich's needs	Rank
1	Fundamental Programming Concepts	3.89	16
2	Fundamental Data Structures	3.81	17
3	Overview of Operating Systems	1.81	20
4	Operating System Principles	2.37	19
5	Concurrency	3.66	18
6	Scheduling and Dispatch	4.74	10
7	Memory Management	5.25	8
8	Resource Allocation and Scheduling	4.12	15
9	Basic Machine Learning	4.57	13
10	Algorithms and Design	4.88	9
11	Parallelism Fundamentals	5.40	7
12	Parallelism	6.44	1
13	Parallel Communication and Coordination	5.57	5
14	Parallel Decomposition	6.13	3
15	Memory System Organization and Architecture	4.72	12
16	Parallel Architecture	5.49	6
17	Parallel Algorithms, Analysis, and Programming	6.17	2
18	Introduction to Modeling and Simulation	6.09	4
19	Distributed Systems	4.21	14
20	Parallel Performance	4.74	11



(Fig. 4) The Knowledge Elements for HPC education using The Locus for Focus Model



4.2.4 고성능컴퓨팅 교육을 위한 지식요소

Borich 요구도 공식의 상위순위에 있는 지식요소들과 The Locus for Focus모델의 HH분면에 공통으로 속하는 지식요소들을 1순위 군으로 도출하고 공통되지 않고 HH사분면과 HL사분면에 포함된 지식요소들을 2순위로 정하여 <Table 8>에 나타내었다. 1순위 군에는 ‘기본 병렬성’, ‘병렬성’, ‘병렬통신 및 조정’, ‘병렬분할’, ‘병렬 알고리즘, 분석 및 프로그래밍’, ‘모델링 및 시뮬레이션 소개’로 6개의 지식요소가 도출되었으며 2순위 군에는 ‘기본 프로그래밍 개념’, ‘기본 자료 구조’, ‘메모리 관리’, ‘알고리즘 및 설계’로 4개의 지식요소가 도출되었다.

도출된 1순위 및 2순위의 지식요소들은 병렬성과 연관이 많으면서 프로그래밍에 관한 지식요소들이었다. 이는 계산과학자들이 고성능컴퓨팅을 활용하기 위해서는 병렬적인 요소들을 반영하여 프로그램을 개발한다는 의미를 가지고 있다.

<Table 8> Borich Rank & LF Quadrant

Knowledge Elements	Rank	LF Quadrant	Priority
Fundamental Programming Concepts	16	HL	2
Fundamental Data Structures	17	HL	2
Overview of Operating Systems	20	LL	
Operating System Principles	19	LL	
Concurrency	18	LL	
Scheduling and Dispatch	10	LH	
Memory Management	8	HH	2
Resource Allocation and Scheduling	15	LL	
Basic Machine Learning	13	LH	
Algorithms and Design	9	HL	2
Parallelism Fundamentals	7	HH	1
Parallelism	1	HH	1
Parallel Communication and Coordination	5	HH	1
Parallel Decomposition	3	HH	1
Memory System Organization and Architecture	12	LH	
Parallel Architecture	6	LH	
Parallel Algorithms, Analysis, and Programming	2	HH	1
Introduction to Modeling and Simulation	4	HH	1
Distributed Systems	14	LH	
Parallel Performance	11	LH	

5. 결론 및 제언

본 연구는 고성능컴퓨팅 전문가와 사용자들을 대상으로 교육요구분석을 통해 계산과학분야의 고성능컴퓨팅 교육을 위한 지식요소들을 제시하였다. 대부분의 관련 연구에서는 고성능컴퓨팅 교육에 대해 단순한 설문조사의 기본적인 통계분석을 통해 지식요소들을 제시하거나, 소수의 전문가 그룹 회의를 통해 도출하였다. 또한 일부 연구에서는 도출된 요소들이 지식요소와 기술요소가 혼용되어 있으며 지식요소들을 도출하기 위한 방법이 자세히 설명되지 않았다. 그리고 도출된 지식요소에 대해 우선순위가 정해지지 않아 교육과정을 개발하고 활용하는데 일부 제한점을 가지고 있었다. 이와 달리 본 연구는 미국 컴퓨터과학의 커리큘럼인 CS2013을 활용하여 고성능컴퓨팅 교육을 위해 전문가 및 사용자들의 설문 자료에 대한 교육 요구를 분석하여 1순위 지식요소 6개와 2순위 지식요소 4개를 고성능컴퓨팅 교육을 위한 지식요소들을 도출하였다.

본 연구의 결과가 의미하는 시사점은 2가지이다. 첫째, 가장 중요하면서 성취도가 낮은 1순위에 포함된 지식요소들은 CS2013의 Parallel and Distributed Computing 지식영역에 속하고 나머지 1개가 Computational Science에 속하였다. 이러한 결과는 현재 사용자들은 병렬성과 연관된 지식요소들에 대해 ‘왜 아직도 성취도가 낮을까’ 하는 것이다. 이는 설문대상자들의 50% 이상이 6년 이상의 활용경력과 최종학력의 56%가 박사인 모집단에서 나온 결과라서 시사하는 바가 크다. 특히 국내 고성능컴퓨팅 교육과정중에서 병렬성에 관해서는 사용자가 쉽게 습득할 수 없을 정도로 매우 열악하여 고성능컴퓨팅에 관한 기본 및 전문교육이 제공되는 것이 쉽지 않음을 보여주고 있다. 국내의 고성능컴퓨팅 교육 중에서도 병렬성에 대한 교육환경을 조성하기 위하여 학부 교육과정과 대학원 교육과정에 병렬성과 관련된 기본교육과정과 전문교육과정을 분리하여 운영하되 연속성을 가질 수 있는 교육과정을 개발하여 병렬성에 대한 여러 전문지식들을 배울 수 있는 체계가 마련되어야 한다.

둘째, 2순위로 정해진 지식요소들은 대부분 소프트웨어개발과 관련된 지식요소들이다. ‘기본프로그래밍’, ‘기본 자료구조’, ‘알고리즘과 설계’ 3가지 지식요소들은 소

프트웨어를 개발할 때 밀접한 관계를 가지고 있다. 고성능컴퓨팅 활용을 위해서 소프트웨어를 개발하는 역량이 중요하며, 사용자들이 소프트웨어를 개발하고 적용하는 경우가 많다는 것을 나타내고 있다. 또한 교육요구도 분석을 통해, 중요하지만 성취도가 높게 나온 지식요소들은 고성능컴퓨팅을 활용하기 위해서 기본적인 주제들로부터 습득해야할 지식요소들을 의미한다. 따라서 기초 또는 기본교육과정에 꼭 2순위로 정해진 지식요소들이 공통적인 교육과정에 포함되어야 한다.

본 연구에서는 일부 제한점을 가지고 있다. 일부 고성능컴퓨팅 교육 요구도에 따라 우선순위가 정해졌지만, 각분야의 설문 표본 부족으로 사용자들의 활용분야인 물리, 화학, 생명과학, 지구과학에 따라 교육요구분석을 수행하지 못하였다. 또한 같은 이유로, 사용자들의 활용 수준에 따라 교육요구분석을 못하였기에 본 연구를 기반으로 향후 후속 연구의 방향을 제안하면 다음과 같다.

첫째, 사용자들의 활용분야에 따른 교육요구를 분석한 결과를 바탕으로 분야별 고성능컴퓨팅 교육과정 개발이 필요하다. 둘째, 사용자들의 활용 수준 진단을 통한 교육요구 분석과 이를 적용한 교육과정 개발이 필요하다. 이를 위해서는 사용자들의 고성능컴퓨팅 활용 수준을 진단하기 위한 도구 개발과 검증에 관한 연구가 필요하며, 이를 기반으로 사용자 수준별 교육요구 분석 결과를 바탕으로 교육과정을 개발하는 연구가 지속되어야 한다.

본 연구에서 제안된 10개 지식요소들은 계산과학분야와 다른 분야에 융합되어 고성능컴퓨팅 교육을 위한 기초자료로 활용할 것을 제안한다.

### 참고문헌

- [1] Stevenson, D. E. (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.
- [2] Education, S. W. G. O. C. (2001). Graduate education in computational science and engineering. *SIAM Review*, 43(1), 163-177.
- [3] Reed, D. A., Bajcsy, R., Fernandez, M. A., Griffiths, J. M., Mott, R. D., Dongarra, J., ... & Ponick, T. L. (2005). Computational science: Ensuring America's competitiveness. PRESIDENT'S INFORMATION TECHNOLOGY ADVISORY COMMITTEE ARLINGTON VA.
- [4] Bo, L., Zhenliu, Z., & Xiangfeng, W. (2012, March). A survey of HPC Development. In *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on* (Vol. 2, pp. 103-106). IEEE.
- [5] National Academies of Sciences, Engineering, and Medicine. (2016). Future directions for NSF advanced computing infrastructure to support US science and engineering in 2017-2020. National Academies Press.
- [6] P.L. 102-94 The High Performance Computing Act of 1991
- [7] ACT ON UTILIZATION AND FOSTERING OF NATIONAL SUPER-COMPUTERS, Republic of Korea(Jul. 26, 2017)
- [8] Van De Vanter, M. L., Post, D. E., & Zosel, M. E. (2005, May). HPC needs a tool strategy. In *Proceedings of the second international workshop on Software engineering for high performance computing system applications* (pp. 55-59). ACM.
- [9] Zarestky, J., & Bangerth, W. (2014, November). Teaching High Performance Computing: Lessons from a flipped classroom, project-based course on finite element methods. In *Proceedings of the Workshop on Education for High-Performance Computing* (pp. 34-41). IEEE Press.
- [10] Yasar, O., Rajasethupathy, K. S., Tuzun, R. E., McCoy, R. A., & Harkin, J. (2000). A new perspective on computational science education. *Computing in Science & Engineering*, 2(5), 74-79.
- [11] Shephard, M. S., Smith, C., & Kolb, J. E. (2013). Bringing hpc to engineering innovation. *Computing in Science & Engineering*, 15(1), 16-25.

- [12] ACM/IEEE-CS Joint Task Force on Computing Curricula, Computer Science Curricula 2013, ACM Press and IEEE Computer Society Press, December 2013
- [13] Borich, G. D. (1980). A needs assessment model for conducting follow-up studies. *Journal of teacher education*, 31(3), 39-42.
- [14] Jo, D. (2009). Exploring How to Set Priority in Need Analysis with Survey. *A Journal of Research in Education*, 35, 165-187.
- [15] Gallopoulos, S., Houstis, E. N., & Rice, J. R. (1992). Future research directions in problem solving environments for computational science.
- [16] Yasar, O., & Landau, R. H. (2003). Elements of computational science and engineering education. *SIAM review*, 45(4), 787-805.
- [17] Fabricius, U., Freundl, C., Köstler, H., & Rüde, U. (2005, May). High performance computing education for students in computational engineering. In *International Conference on Computational Science* (pp. 27-35). Springer, Berlin, Heidelberg.
- [18] Squires, S., Van De Vanter, M., & Votta, L. (2006, February). Yes, There Is an "Expertise Gap" In HPC Applications Development. In *Third Workshop on Productivity and Performance in High-End Computing (PPHEC06)*.
- [19] Armosky, B., Brown, S., Drummond, T., Ferguson, J., Gerber, R., Hacker, T. J., ... & Traxler, K. (2007). HPC University. In *TG08-TeraGrid Conference, Madison, WI*.
- [20] EDUCATION, S. W. G. O. C. U., Co-Chairs, P. T. A. L. P., Shiflet, A., Vakalis, I., Jordan, K., & John, S. S. (2011). Undergraduate computational science and engineering education. *SIAM review*, 53(3), 561-574.
- [21] Sancho, M. R., Alexandrova, N., & Gonzalez, M. (2015, December). Addressing HPC skills shortages with parallel computing MOOC. In *Interactive Collaborative and Blended Learning (ICBL), 2015 International Conference on* (pp. 86-93). IEEE.
- [22] Wilson, L. A., & Dey, S. C. (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.
- [23] Gordon, S. I., Demmel, J., Destefano, L., & Rivera, L. (2016). Implementing a Collaborative Online Course to Extend Access to HPC Skills. *Computing in Science & Engineering*, 18(1), 73-79.
- [24] Cahill, K. J., Lathrop, S., & Gordon, S. (2017). Building a Community of Practice to Prepare the HPC Workforce. *Procedia computer science*, 108, 2131-2140
- [25] Rüde, U., Willcox, K., McInnes, L. C., & Sterck, H. D. (2018). Research and education in computational science and engineering. *Siam Review*, 60(3), 707-754.
- [26] SIAM, <https://www.siam.org/>
- [27] Park SunJu. (2017). Analysis of Learners' Preferences in SW Education Contents Development. *Journal of the Korean Association of Information Education*, 21(6), 691-699.
- [28] Han Young Shin. (2018). Analysis of Effectiveness of Programming Learning for Non-science Major Preliminary Teachers' Development of Computational Thinking. *Journal of the Korean Association of Information Education*, 22(1), 41-52.
- [29] Lawshe, C. H. (1975). A quantitative approach to content validity 1. *Personnel psychology*, 28(4), 563-575.
- [30] Ayre, C., & Scally, A. J. (2014). Critical values for Lawshe's content validity ratio: revisiting the original methods of calculation. *Measurement and Evaluation in Counseling and Development*, 47(1), 79-86.
- [31] Lee, J. E., & Kim, H. (2016). Analysis of College Students' Educational Needs for Career Education. *Journal of Learner-Centered Curriculum and Instruction*, 16, 1001-1027.

저자소개



**윤 희 준**

2012~현재 성균관대학교 교과교육학과 박사과정

2000~현재 한국과학기술정보연구원

관심분야 : 컴퓨터 교육, HPC, HPC교육, SW교육

e-mail : k2@kisti.re.kr



**안 성 진**

1988 성균관대학교 정보공학과 (학사)

1990 성균관대학교 정보공학과 (석사)

1998 성균관대학교 정보공학과 (박사)

1996 정보통신기술사

1999~현재 성균관대학교 컴퓨터교육과 교수

관심분야 : 컴퓨터 교육, SW교육, 정보윤리, 정보보안

E-Mail : sjahn@skku.edu