Journal of The Korea Society of Computer and Information
Vol. 24 No. 1, pp. 1-7, January 2019
www.ksci.re.kr
https://doi.org/10.9708/jksci.2019.24.01.001

Implementation of a Wi-Fi Based Cluster System using Raspberry Pi for Multidisciplinary Education

Geum-Seo Koo*, Gab-Sig Sim*

Abstract

In this paper, we implemented a Wi-Fi based cluster system using raspberry pi for multidisciplinary education. The cluster implementation on the desktop was more difficult to maintain the complexity, big size, high price, power consumption as the number of nodes increased. In this paper, we implemented a cluster using Raspberry Pi, which is developed for educational purposes, to reduce the cost of connecting nodes. In addition, the complexity of system construction is reduced by replacing the connection between each node with Wi-Fi. Also, the inconvenience of configuration due to node increase was reduced. It is expected that the implementation of the cluster will be a good alternative in the educational environment where distributed processing and parallel processing are performed in the embedded environment. Also, it is confirmed that it can be applied to the multidisciplinary education.

▸Keyword: Raspberry Pi, Cluster, MPICH, IoT, Parallel Programming, Multidisciplinary Education

I. Introduction

사물인터넷(Internet of Things, IoT)[1]과 빅데이터(Big data) 그리고 인공지능(Artificial Intelligence, AI)으로 상징되는 4차 산업혁명시대에서 처리해야할 데이터 또한 기하급수적으로 증가하고 있다. IoT 환경에서 방대한 데이터 처리에 부족함이 발생하면 이를 극복하기 위해서 한 대의 기기가 아닌 여러 IT기기를 묶어서 병렬 처리하는 클러스터(Cluster)[2-4] 기술이 효율적이다. 클러스터 구축으로 병렬 프로그래밍 및 병렬 처리를 하기 위해서는 MPI(Message Passing Interface)를 사용한다. 최근 멀티 코어 등의 등장으로 병렬 처리의 필요성이 높아졌으며, 이로인해 운영체제 및 병렬 프로그래밍 등의 전공과목에서 관련 내용의 학습과 실제 환경을 구축해 보는 과정이 중요하다. 하지만 일부 전공과목에서 필요한 클러스터를 구축하기 위해서는 최소 2개 이상의 데스크탑과 이를 연결하기 위한 네트워크 환경이 필요하며 이를 구축하기 위한 비용과 시간등에 어려움이 많이 발생한다. 그러므로 임베디드 등의 이름으로 개설되는 과목에서 많이 사용하는 장비인 라즈베리 파이를 활용하여 전공 과목간의 연계성을 높여 보고자 한다. IoT 및 임베디드 관련 분야에서 많이 사용되는 대표적인 교육용 보드는 영국에서 개발된 라즈베리 파이(Raspberry Pi)[5]이다. 라즈베리 파이는 컴퓨팅 파워가 일반 데스크탑에 비해 부족하지만 초소형에 저렴한 비용으로 사용할 수 있으며, 기본 운영체제로 리눅스를 사용하고 있으므로 활용성이 매우 높다. 본 연구에서는 교육목적으로 많이 사용되고 있는 초소형 컴퓨터인 라즈베리 파이 3 Model B의 Wi-Fi 기능을 활용하여 클러스터 환경을 구축하는 방법을 제시한다. 또한 이를 활용하여 융복합 교육 및 각 전공과목간의 연계성을 높일 수 있다. 시간이 지날수록 높은 컴퓨팅 능력이 요구되고 있는 상황에서 Wi-Fi를 활용한 효율적 환경과 구축이 좋은 대안이 될 수 있을 것으로 기대한다.

• First Author: Geum-Seo Koo, Corresponding Author: Gab-Sig Sim
*Geum-Seo Koo (goodman4009@gmail.com), Department of Liberal Arts, Gyeongnam National University of Science and Technology
*Gab-Sig Sim (gssim@gntech.ac.kr), Department of Liberal Arts, Gyeongnam National University of Science and Technology
• Received: 2018. 10. 16, Revised: 2018. 11. 30, Accepted: 2018. 12. 09.
• This work was supported by Gyeongnam National University of Science and Technology Grant in 2018~2019.

II. Preliminaries

1. Related works

사물인터넷 시대의 도래로 데이터 증가 및 처리 능력의 향상이 중요하다. 이를 위해서 최근 멀티코어 및 클러스터를 활용한 병렬 처리와 관련된 연구 및 교육이 증가하고 있다. IoT 시대에 대응하기 위해서 대학의 ICT관련 학과에서는 관련 과목(IoT Embedded Programming, Parallel Programming, Linux System, 운영체제 일부 내용 등)이 많이 개설되어 운영 중이며, IoT 교육에서 많이 사용되고 있는 보드로는 라즈베리 파이와 아두이노(Arduino)[6], ATmega128, 8051 등이 있다. 라즈베리 파이는 영국 잉글랜드의 라즈베리 파이 재단에서 교육 목적으로 개발된 싱글 보드 컴퓨터(Single Board Computer, SBC)이며 가격이 저렴해서 사용 시 부담이 적다. 또한 초소형 컴퓨터지만 일반 컴퓨터가 갖춰야하는 최소 구성 요소를 모두 갖추고 있으며 보드를 구동하는 기본 운영체제는 리눅스 기반의 라즈비안(RASPBIAN)으로 활용성이 매우 높다. 이러한 특징과 활용성으로 라즈베리 파이를 이용한 다양한 연구[7-9]가 진행되고 있다.

라즈베리 파이가 많은 장점을 가지고 있지만 데스크탑과 비교되는 성능으로 많은 양의 데이터 처리에는 한계가 발생하므로 클러스터로 구축하면 효율적으로 결과를 도출 할 수 있다. 기존 데스크탑 기반에서 적용해 온 클러스터 구축은 여러대의 컴퓨터를 네트워크로 연결해서 병렬컴퓨팅을 할 수 있도록 만든 것이다. 이를 위해서 메시지 전달 인터페이스(Message Passing Interface, MPI[10-12]의 병렬 계산 소프트웨어인 MPICH2[13-15] 환경에서 사용한다. MPI는 분산 및 병렬 처리에서 정보의 교환에 관해 기술하는 표준이다. 메시지 패싱(Message Passing) 방식은 프로세서 간에 교환할 데이터를 메시지 전달함수를 사용해 주고받는 연산 모델로서 이런 함수들의 집합체인 메시지 패싱 라이브러리(Message Passing Library)의 표준을 정한 것이 MPI이며 이에 맞추어 여러 MPI 라이브러리가 개발되어 사용중이다. 슈퍼컴퓨터[16]도 유사한 방식으로 만들어진다. 하지만 물리적으로 네트워크를 통해서 처리하므로 빠른 처리를 위해서 네트워크 성능이 중요하며 이를 구축하기 위해서 많은 비용이 소모된다.

본 연구에서는 IoT 임베디드 환경에서 라즈베리 파이를 활용[17-22]하여 Wi-Fi 기반으로 대량의 데이터를 처리할 수 있는 효율적인 Wi-Fi 기반 Cluster 구축 방법을 제시한다. 또한 병렬 컴퓨팅 및 프로그래밍 관련 과목등과의 연계성을 높여 융복합 교육에 활용할 수 있다.

III. The Proposed Scheme

본 제안에서는 라즈베리 파이의 Wi-Fi를 이용하여 클러스터(Cluster) 환경을 구축하는 방법을 제시하며, 전공과목에서 필요로하는 병렬처리, Linux System 및 관련 프로그래밍 과목등과의 연계방법을 제시한다.

1. Implementation of Cluster System

Table 1. Raspberry Pi 3 Model B Spec.

Name	Raspberry Pi 3 Model B Rev 1.2
CPU	Boradcom BCM2387 chip 1.2GHz Quad-Core ARM Cortex-A53 64bit
GPU	Dual Core VideoCore IV, Multimedia Co-Processor, Provides Open GL
Memory	1GB LPDDR2
Memory Slot	Push/pull SDIO
Ethernet/WIFI	10/100 BaseT Ehternet socket
USB 2.0	4 * USB 2.0 Connector
I/O	26 GPIO, 1 Uart, 1 SPI, 2 I2C, PCM/i@s, 2PWM CSI & DSI
GPIO	Connector 40-pin 2.54mm(100mil) expansion header: 2*20 strip Providing 27GPIO pins as well as +3.3V, +5V and GND supply lines
OS	Linux, Android, Windows 10
Power	Micro USB socket 5V1, 2.5A
Price	$35
Dimensions	85 * 56 * 17m

본 절에서는 라즈베리 파이 3 모델 B를 이용하여 Wi-Fi 기반의 효율적인 클러스터(Cluster) 환경을 구축하는 방법을 설명한다. SBC(Single Board Computer) 기본 사용은 [Table. 1.]과 같다. 라즈베리 파이는 데비안 리눅스를 기반으로 만들어진 전용 운영체제인 RASPBIAN(Raspbian Stretch With Desktop, Version: November 2017, Kernel Version: 4.9, Size: 1.6GB)을 사용하여 라즈베리 파이 공식 홈페이지(http://www.raspberrypi.org)에서 다운로드 가능하다. 기본 설치 작업시 NOOBS(New Out Of the Box Software) 기반으로도 설치 가능하다. 운영제체가 담길 저장장치로 SD카드를 사용하므로 SD카드 포맷을 위해서 SD Formatter가 필요하다. OS 이미지 복사를 위한 Win32DiskImager도 설치 작업시 사용된다. 병렬처리를 위해서 MPI(Message Passing Interface)의 MPICH2를 설치 후 라즈베리 파이에 맞게 환경 설정을 한다. 기본적으로 연산에 관여하는 노드(Node)는 최소 3기에 설정하며 최대 32개의 노드에 설치한다. 전체적인 작업 절차는 Raspberry Pi 업데이트를 거친 후, MPICH를 다운로드(http://www.mpich.org/downloads/)한다. 그 다음 컴파일 환경을 만든 후 재부팅 하여 기본 코드 동작을 확인한다. 이를 통해서 기존 데스크탑 기반 환경 대비 Wi-Fi 기반의 구축으로 비용 부담을 줄이고 시스템 구축이 효율적임을 보인다.

전체 먼저 3개의 노드(Node)에 Raspberry Pi가 정상적으로 설치되었다고 가정하고 Master Node를 AP(Access Point)로 만드는 작업을 선행한다. 작업 내용은 shell로 작성하였으며 내용은 다음과 같다.

```
#!/bin/bash
#if [ "$EUID" -ne 0 ]
        then echo "Must be root"
        exit
fi
if [[ $# -lt 1 ]];
        then echo "You need to pass a password!"
        echo "Usage:"
        echo "sudo $0 yourChosenPassword [apName]"
        exit
fi
APPASS="$1"
APSSID="rPi3"
if [[ $# -eq 2 ]]; then
        APSSID=$2
fi
apt-get remove --purge hostapd -yqq
apt-get update -yqq
apt-get upgrade -yqq
apt-get install hostapd dnsmasq -yqq
cat > /etc/dnsmasq.conf <<EOF
interface=wlan0
dhcp-range=10.0.0.2,10.0.0.5,255.255.255.0,12h
EOF
cat > /etc/hostapd/hostapd.conf <<EOF
interface=wlan0
hw_mode=g
channel=10
auth_algs=1
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_pairwise=CCMP
rsn_pairwise=CCMP
wpa_passphrase=$APPASS
ssid=$APSSID
ieee80211n=1
wmm_enabled=1
ht_capab=[HT40][SHORT-GI-20][DSSS_CCK-40]
EOF
sed -i -- 's/allow-hotplug wlan0//g' /etc/network/interfaces
sed -i -- 's/iface wlan0 inet manual//g' /etc/network/interfaces
sed -i --
's/ wpa-conf ₩/etc₩/wpa_supplicant₩/wpa_supplicant.conf//g'
/etc/network/interfaces
sed -i --
's/#DAEMON_CONF=""/DAEMON_CONF="₩/etc₩/hostapd₩/ho
stapd.conf"/g' /etc/default/hostapd
cat >> /etc/network/interfaces <<EOF
# Added by rPi Access Point Setup
allow-hotplug wlan0
iface wlan0 inet static
        address 10.0.0.1
        netmask 255.255.255.0
        network 10.0.0.0
        broadcast 10.0.0.255
EOF
echo "denyinterfaces wlan0" >> /etc/dhcpcd.conf
systemctl enable hostapd
systemctl enable dnsmasq
sudo service hostapd start
sudo service dnsmasq start
echo "All done! Please reboot“
```

Fig. 1. Shell Script of Master Node

다음으로 라즈베리 파이에 Cluster 구축을 위한 기본적인 작업 절차는 다음과 같이 진행한다. 아래의 절차는 일반적인 Cluster 구축 방법이므로 상황에 따라 일부 내용을 변경하여 설치한다. 먼저 “sudo raspi-config” 명령어를 실행하여 [Fig. 1.]과 같은 설정화면에서 원하는 hostname으로 바꾼다. 그리고 SSH(Secure Shell)은 활성으로 변경한다. 이곳에서 ID(pi)와 Password(raspberry)를 변경할 수 있다. 설정 이후에 동일한 네트워크에서 라즈베리 파이에 접속한다. Windows에서는 putty를 통해 IP 주소로 접속하여, Linux 계열에서는 터미널에서 ssh pi@ipaddress로 접속한다.

Cluster 작업시 병렬처리는 MPICH 3.1을 사용했으며 설정 절차는 아래 내용과 같다. MPICH 설치 및 설정하는 방법을 간략하게 요약하면 먼저 전체 업데이트 이후에 MPICH2 디렉토리에 wget 명령어로 MPICH를 다운로드 받는다. 다운로드 받은 “mpich-3.1.tar.gz”은 Size가 대략 10.5MB 정도이며 tar로 묶인 뒤 gz으로 압축된 상태다. 이 파일을 압축 해제하고 컴파일 및 설치를 진행한다.

```
$ sudo apt-get update
$ mkdir mpich2
$ cd ~/mpich2
$ wget http://www.mpich.org/static/downloads/3.1/mpi
  ch-3.1.tar.gz
$ tar xfz mpich-3.1.tar.gz
$ sudo mkdir /home/rpimpi/
$ sudo mkdir /home/rpimpi/mpi-install
$ mkdir /home/pi/mpi-build
$ cd /home/pi/mpi-build
$ sudo apt-get install gfortran
$ sudo /home/pi/mpich2/mpich-3.1/configure-prefix=/
  home/rpimpi/mpi-install
$ sudo make
$ sudo make install
$ cd ..
$ nano .bashrc
$ sudo reboot
```

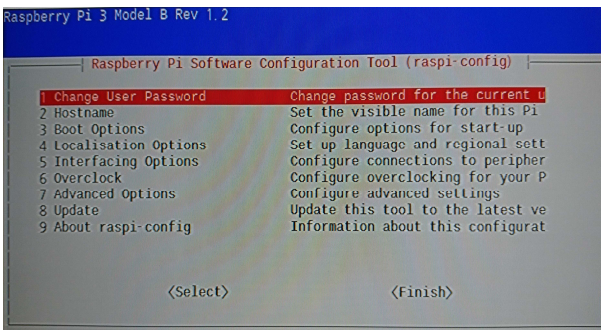

Fig. 2. Raspi-config Environment

라즈베리 파이상에서 MPICH 설치 작업을 진행하면 [Fig. 2.]와 같이 컴파일 완료에만 대략 30분 전후의 시간이 소요되며 전체 작업을 완료하는데 1시간 가까운 시간이 필요하다. 작업 마지막 명령어 중 “nano .bashrc” 파일의 가장 아래에 “PATH=$PATH: /home/rpimpi/mpi-install/bin”을 추가하고 시스템을 재부팅한다. 위 작업으로 MPICH 기본 설치는 마무리

되었으며 마지막으로 라즈베리 파이의 기본 언어인 Python이 MPICH에서 동작하도록 설정하는 작업은 아래와 같다.

```
$ sudo aptitude install python-dev
$ wget https://bitbucket.org/mpi4py/mpi4py/downloads/
  mpi4py-2.0.0.tar.gz
$ tar -zxf mpi4py-1.3.1.tar.gz
$ cd mpi4py-1.3.1
$ python setup.py build
$ sudo python setup.py install
$ export PYTHONPATH=/home/pi/mpi4py-1.3.1
$ mpiexec -n 5 python demo/helloworld.py
```

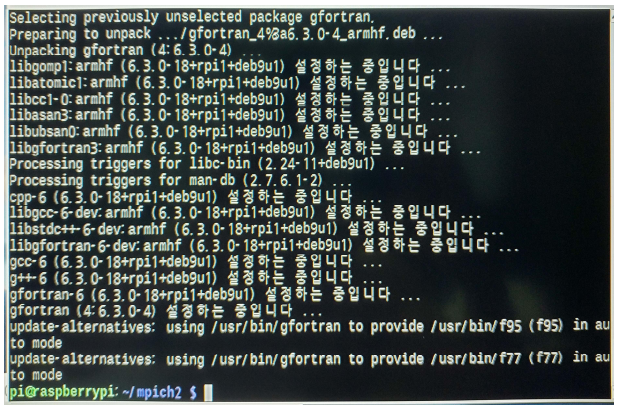

Fig. 3. MPICH Compile

Python Code가 MPICH에서 정상적으로 동작하는 확인한 결과는 [Fig. 3.]과 같다. 지금까지의 과정을 살펴보면 라즈베리 파이, MPICH, Python 설치 및 설정 과정에 적지 않은 시간이 소요된다. Cluster 환경에서는 모든 노드(Node)에 동일한 환경이 필요하므로 반복적 작업과 시간낭비가 일어나므로 위 작업 내용을 “.img” 파일로 압축을 수행하여 “diskimager”를 통해서 “name.img”로 read 하면 많은 시간을 줄일 수 있다. 다음 노드(Node)는 “write” 명령어로 간단히 환경을 구축 할 수 있다. 각각의 노드에 기본 작업 내용 복사가 끝나면 Cluster에 사용할 모든 라즈베리 파이를 연동시키는 작업을 수행한다. 먼저 서버의 마스터 역할을 수행하는 nmap으로 서버(Server)로 동작될 라즈베리 파이에 원격으로 접속하고 아래의 명령을 수행한다.

```
$ sudo apt-get update
$ sudo apt-get install nmap
$ sudo nmap -sn 192.168.1.*
```

연결된 모든 라즈베리 파이의 IP를 확인하고, MPICH 동작을 위해서 아래와 같이 IP입력 파일을 만들고 “machinefile”에 기록한다.

```
$ mkdir mpi_test
$ cd mpi_test
$ nano machinefile
```

작업의 편의를 위해 모든 라즈베리 파이 이름을 변경하기 위해서 “sudo raspi-config” 명령어을 수행한다. 이후 각 라즈베리 파이의 연동을 위해서 암호를 적용하고 노드간 접속을 자동화하며 설정은 아래와 같다.

```
허브서버 pi01에 원격 접속
$ ssh-keygen
$ cd ~
$ cd .ssh
$ cp id_rsa.pub pi01

- 노드 pi0에 원격 접속
$ ssh-keygen
$ cd .ssh
$ cp id_rsa.pub pi
$ scp 허브인PI01의IP:/home/pi/.ssh/pi01 .
$ cat pi01 >> authorized_keys
$ exit
```

- 허브서버인 pi01 에서 나머지 모든 라즈베리 파이를 수용하는 작업은 아래와 같다.

```
$ scp PI0?의IP:/home/pi/.ssh/pi0? .
$ cat pi0? >> authorized_keys
$ mpiexec -f machinefile -n 3 hostname
```

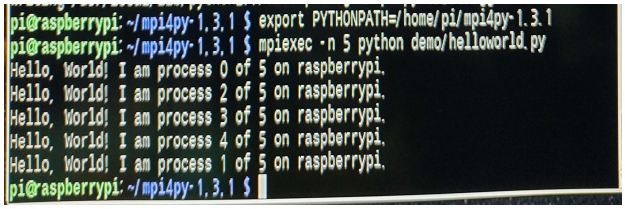

Fig. 4. Python Operation

Fig. 5. Wi-Fi Based Raspberry Pi Cluster

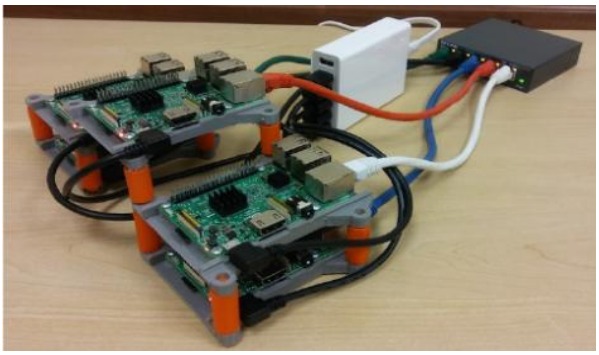

Fig. 6. Ethernet Based Raspberry Pi Cluster

Table 2. The Comparisons of Ethernet Based Pi and Wi-Fi Based Pi Cluster

Item	Ethernet Based Pi	Wi-Fi Based Pi
Connectivity	Normal	Good
Scalability	Normal	Good
Space Occupancy	Normal	Lowness
Power Consumption	Normal(hub, node)	Lowness(node)
Cost	Normal(hub, node)	Lowness(node)
Performance	Normal	Normal

라즈베리 파이 Cluster의 연결 상태 테스트를 수행하여 정상적으로 동작하는지 확인한다. [Fig. 4.]는 최종적으로 구현된 3 Node 라즈베리 파이이며 Wi-Fi 기반의 Cluster이다. [Fig. 5.]는 기존의 Ethernet 기반의 라즈베리 파이 Cluster[25]이다. [Fig. 5.]와 같이 Cluster를 Ethernet 기반으로 구축하면 Node 증가에 따른 물리적 연결의 복잡함이 증가되며 공간 활용에 제한이 발생한다. 또한 Hub의 연결 가능한 최대 Port에 제한이 발생하여 최대 Port를 넘어서 연결할 때 추가 Hub가 필요하게 되며 이에 따른 복잡도, 공간 효율성, 전력 소모 등에서 불리하다. 본 논문에서 제시한 Wi-Fi 기반 Cluster는 Node 증가에 따른 물리적인 Ethernet 연결이 불필요하며 간단한 설정만으로 Cluster 추가 구성이 가능하여 유연한 구축이 가능하다. 또한 공간상의 복잡성이 줄어든다. Wi-Fi 기반의 라즈베리 파이 Cluster는 저비용, 작은 크기, 낮은 전력소모, 복잡성 감소 등의 장점을 얻을 수 있었다. 기존의 Ethernet 기반의 라즈베리 파이 Cluster와 Wi-Fi 기반의 라즈베리 파이 Cluster를 비교한 내용은 [Table 2.]와 같다. 표와 같이 Node의 물리적 연결에서 기존 시스템 대비 편리하며, 여러 Node 추가 등으로 확장시에도 Wi-Fi 기반이므로 상대적으로 편리함을 확인할 수 있다. 또한 복잡한 케이블링이 불필요하므로 공간 효율이 높으며, 부가적으로 Hub 미사용으로 전력 소비 및 비용도 줄어든다. 라즈베리 파이 기반의 Cluster는 데스크탑 대비 낮은 컴퓨팅 능력으로 대량의 데이터 처리에 한계가 보이지만 노드의 추가로 성능 향상을 추구할 수 있다. 이와같은 시스템의 구축으로 임베디드 환경에서의 분산 처리 및 병렬처리를 진행하는 교육 환경에서는 좋은 대안이 될 수 있다.

2. Multidisciplinary Education

4차 산업혁명 및 IoT는 대학 교육에서 중요한 분야이므로 관련 과목의 개설이 많이 증가하고 있다. 본 연구에서는 IoT 관련 교과목인 “IoT Embedded Raspberry Pi Programming”에서 사용하는 라즈베리 파이와 운영체제, 리눅스 시스템 및 프로그래밍 언어 등의 과목에 본 Cluster 시스템을 사용하여 과목간 연계에 활용 할 수 있음을 확인하였다.

1) IoT Embedded Raspberry Pi Programming

사물인터넷 관련 기반 기술을 학습하기 위해서는 기본 이론 및 보드 실습은 필수적이다. 이론 교육은 라즈베리 파이의 소개와 특징 및 IoT, Embedded, Python, Linux 등의 내용을 학습한다. 실습은 주제별 혹은 프로젝트 단위로 다양한 예제(LED, Motor 구동 등)를 구현한다. 추가적으로 타 전공과목 연계를 위해서 본 논문에서 제시한 Cluster 구축 방법을 학기 후반부에 학습 할 수 있다. 또한 라즈베리 파이 Cluster 기반으로 병렬 처리를 위한 기본적인 MPICH도 학습 가능하다. 이로 인해서 동일 학기에 동시에 진행되는 운영체제 과목의 후반부에 진행되는 분산 및 병렬 처리 시스템 영역에서 실제 예제를 기반으로 학습을 효율적으로 할 수 있다.

2) Operating System

운영체제는 운영체제의 개념, 구조, 기능 등 전반적인 동작 과정을 교육하지만 분산 및 병렬 처리 시스템은 환경 구축의 어려움으로 생략하거나 이론으로 마치는 경우가 대부분이다. 하지만 본 논문에서 제시한 Cluster 시스템으로 실제 학습에 적용한 예는 다음과 같다.

운영체제의 학습 내용 중 Deadlock은 2개 이상의 프로세스들이 블록킹 되어 각각 다른 프로세스의 진행을 기다릴 때 발생하며 이는 중요한 오류 중 하나이다. 다음 예제는 [Fig. 7.] Deadlock에 대한 예제이다.

```
#include <stdio.h>
#include <mpi.h>
void main(int argc, char * argv[]) {
  int nprocs, myrank;
  MPI_Status status;
  double a[100], b[100];
  MPI_Init(&argc, &argv);
  MPI_Comm_size(MPI_COMM_WORLD, &nprocs);
  if(myrank == 0) {
    MPI_Recv(b, 100, MPI_DOUBLE, 1, 19,
                    MPI_COMM_WORLD, status);
    MPI_Send(a, 100, MPI_DOUBLE, 1, 17,
                    MPI_COMM_WORLD);
  }else if(myrank == 1) {
    MPI_Recv(b, 100, MPI_DOUBLE, 0, 17,
                    MPI_COMM_WORLD, status);
    MPI_Send(a, 100, MPI_DOUBLE, 0, 19,
                    MPI_COMM_WORLD);
  }
  MPI_Finalize();
}
```

Fig. 7. Deadlock Example

MPI의 Deadlock 예제를 간략히 설명하면 각각의 프로세스는 서로에 의해 송신되는 메시지를 수신하면서 시작되는데 프로세스 0은 프로세스 1이 메시지를 보내야 진행이 되며, 프로세스 1은 역시 프로세스 0이 메시지를 보내야 진행이 된다. 이때 아무런 메시지도 송신되지 않고 따라서 아무런 메시지도 수신되지 않는다. MPI의 내부 버퍼의 크기에 의존하는 프로그램은 이식성과 확장성 측면에서 바람직하지 않으며 배열의 크기를 MPI 내부 버퍼보다 크게 실행한다면 대부분의 시스템에서 Deadlock에 빠짐을 확인 할 수 있다.

위의 코드와 같이 운영체제에서 이론적으로 학습하는 Deadlock을 실제 Cluster에서 간단히 학습이 가능하다. 또한 데스크탑 기반이 아닌 라즈베리 파이 기반이므로 공간 및 네트워크 연결에 따른 제약이 현저히 줄어 든다. 부가적으로 라즈베리 파이 기반의 Cluster 환경은 기본 운영체제로 Linux 및 Python을 사용하므로 관련 과목에서도 유용하게 적용할 수 있음을 확인하였다.

본 논문에서 제시한 Cluster로 시스템 구축에 따른 비용 및 Wi-Fi의 사용으로 노드 증가에 따른 네트워크 환경의 복잡성이 현저히 줄어 들었으며, 다양한 교과목 연계를 통한 융복합 교육 환경 개선에 활용성이 높음을 확인하였다.

IV. Conclusions

제4차 산업혁명 시대의 문턱에서 ICT 분야는 빠르게 변화하고 있다. IT 기기는 갈수록 고성능 및 소형화 되고 있고 더욱더 많은 데이터를 처리하고 있다. 이러한 환경에서 대량의 데이터를 처리할 수 있는 Wi-Fi 기반의 클러스터는 활용분야가 다양하다. 일반 PC와 같이 리눅스 기반의 클러스터 환경은 슈퍼컴퓨터와 유사하게 빠른 계산이 필요한 부분에 사용될 수 있으며 기존 모델 대비 연결성, 크기, 공간 활용, 전력소모 및 가격 등에서 유리하다. 또한 본 논문에서 제시한 시스템으로 노드 증가에 따른 네트워크 시스템의 복잡성이 현저히 줄어 들었으며, 이로 인해서 컴퓨터공학 관련 전공 과목간의 연계 및 융복합 교육에 효율적임을 확인하였다. 라즈베리 파이 자체의 컴퓨팅 능력도 높아지고 있으므로 Wi-Fi 기반 클러스터 환경으로 더욱 강력하고 효율적인 컴퓨팅 환경이 제공되길 기대해 본다. 또한 라즈베리 파이를 이용한 슈퍼컴퓨터 및 스마트 폰 등이 등장하는 현 시점에서 라즈베리 파이를 교육에 효율적으로 적용할 수 있는 더 적극적인 연구가 필요할 것으로 생각한다. 그러므로 추후 연구로는 Linpack[23, 24]을 활용한 데스크탑 환경과 비교하여 실제 실무에 적용 가능한 분야를 확인하고, 라즈베리 파이를 활용한 유사 과목간에 최적화되고 구체적인 커리큘럼을 개발하는 연구 및 구현을 진행할 것이다.

REFERENCES

[1] D. Bradley, D. Russel, I. Ferguson, J. Isaacs, A. MacLeod, and R. White, “The internet of things the future or the end dof mechatronics, ”Mechatronics, vol. 27. pp. 57-74, 2015.

[2] W. Gropp, E. Lusk and Thomas, “*Beowulf Cluster Computing with Linux, Second Edition.*” 2003.

[3] M. A. Baker, G. C. Fox, and H. W. Yau, “Cluster Computing Review”, NPAC Technical Report SCCS-748, 1995.

[4] Simon J. Cox, “Iridis-pi: a low-cost, compact demonstration cluster”, 2013

[5] Raspberry Pi, *https://www.raspberrypi.org*

[6] ARDUINO, *https://www.arduino.cc*

[7] S. Ferdoush and X. Li, “Wireless Sensor Network System Design using Raspberry Pi and Arduino for Emvironmental Monitoring Applications,” Proceedia Computer Science, vol. 34, pp. 1-3-110, 2014.

[8] Steps th make a Raspberry Pi Supercomputer, *http://www.southampton.ac.uk/~sjc/raspberrypi/pi_supercomputer_southampton_web.pdf*

[9] HPL(High Performance Linpack) : Benchmarking Raspberry PIs, *https://www.howtoforge.com/tutorial/hpl-high-performance-linpack-benchmark-raspberry-pi/*

[10] MPI, *http://www.mpi-forum.org*

[11] W. Gropp, E. Lusk, and A. Skjellum. “Using MPI:Portable Parallel Programming with the Message Passing Interface”. MIT Press, 1995.

[12] I. Foster, J. Geister, W. Gropp, N. Karonis, E. Lusk, G. Thiruvathukal, and S. Tuecke. “A wide-area implementatin of the Message Passing Interface.” Parallel Computing, pp. 1735-1749, 1998.

[13] MPICH, *http://www.mpch.org*

[14] N. Karonis, B. Toonen, I Foster, “MPICH-G2: a Grid-enabled implementation of the Message Passing Interface”, Journal of Parallel and Distributed Computing, Volume 63, pp.551-563, 1998.

[15] M. Muller, M. Hess, E. Gabriel, “Grid enabled MPI solutions for Clusters”, In 3rd International Symposium on Cluster Computing and the Grid, pp.18-25, 2003.

[16] TOP 500 The List, *https://www.top500.org*

[17] S. Monk, Programmming the Raspberry Pi Getting started with Python, Mc Graw Hill, 2013.

[18] S. Monk, Programming the Raspberry Pi Cookbook, Hanbit Media, 2015.

[19] D. Norris, Raspberry Pi with Project, Hanbit media, 2015.

[20] Build a Compact 4 Node Raspberry Pi Cluster, *https://makezine.com/projects/build-a-compact-4-node-raspberry-pi-cluster/*

[21] Make your Own Cluster Computer(part1), http://www.tinkernut.com/2014/04/make-cluster-computer/
[22] Make Your Own Cluster Computer(Part2), http://www.tinkernut.com/2014/05/make-cluster-computer-part-2/
[23] Dongarra, J. J.: Luszczek, P. and Petitet, A. "The LINPACK Benchmark; past, present and future," Concurrency and Computation: Practice & Experience, Vol.12, No.9, 2003.
[24] Dongarra, J. J, "LINPACK: user' guide," Society for Industrial and Applied Mathematics, 1979.
[25] Kevin. D. and Jian Z., "Learning Cluster Computing by Creating a Raspberry Pi Cluster", WOODSTOCK'97, July 2016.

Authors

Geum-Seo Koo received the B.S. and M.S. degrees in Computer Science from Gyeongsang National University, Jinju, Korea, in 2003, and 2005, respectively. He is currently a Ph.D candidate in Computer Science at Gyeongsang National University. Jinju, Korea. He is currently interested in Embedded System, System Security and High Performance Computing.

Gab-Sig Sim received the B.S., M.S. and Ph.D. degrees in Computer Science and Statistics from Chonnam National University, Gwangju, Korea, in 1985, 1987 and 1993, respectively. Dr. Sim joined the faculty of the Department of Liberal Arts at Jinju National University, Jinju, Korea, in 1993. He is currently a Professor in the Department of Liberal Arts, Gyeongnam National University of Science and Technology. He is interested in Information Communication and Security, Atrtificial Intelligence Ethics, and Science, Technology and Society.