

IoT 장치를 이용한 Redis, Fabric 상호 협력 기반 기술 연구

오승민*, ASHIQUZZAMAN AKM*, 이동수*, 엄태원** and 김진술*,

A Study on the Technology Based on Redis and Fabric Parallel Processing with IoT Devices

Oh SeungMin*, ASHIQUZZAMAN AKM*, Lee DongSu*, Um TaeWon** and Kim JinSul*

요 약

우리의 생활에 점점 IoT와 빅데이터, 머신러닝을 통한 인공지능 기술들이 접목되어 지고 있다. 이런 IT기술이 발전할 수 있었던 것은 하드웨어적인 발전이 크다. 개발되어진 하드웨어 중 하나인 라즈베리파이는 우리가 효율적으로 교육하고 개발할 수 있게 만들어진 싱글 보드 컴퓨터이다. 우리는 이 라즈베리파이를 이용하여 성능을 끌어올리기위해 클러스터링과 병렬처리에 대한 기술에 대해 알아본다. 이 기술을 통한 구현을 해보며 각 기술에 대한 장단점에 대해 설명한다. 각 기술에 대한 문제점과 문제점을 해결할 시 우리의 생활과 교육에 있을 발전에 대해 이야기한다.

Abstract

Nowadays, our lives have become increasingly integrated with artificial intelligence technologies through IoT, big data and machine learning. One of the these developed hardware, the single-board computer RaspberryPi, has enabled us to develop several application based in the mobility and sparsity. We use this Raspberypai to learn about the technology of clustering and parallelism to boost performance. Also, we focused in implementation of this technology and explain the pros and cons of each technology. later we discussed the disadvantages of each technology and the progress that will be made in our both out daily lives and industrial sector.

Key words

Raspberypai, IoT, Parallel, Parallel Computing, Cluster, MPI, Redis

* 전남대학교 공과대학 전자컴퓨터공학과 ** 조선대학교 IT융합대학 정보통신학부

※ 2018년도 과학기술정보통신부의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(2018-0-00691, 초소형 IoT 디바이스를 위한 자율적 상호협력기반 군집기능 기술개발). 게다가, 본 논문은 중소벤처기업부에서 지원하는 2018년도 산학연협력 기술개발사업(No. S2655639)의 연구수행으로 인한 결과물임을 밝힙니다.

I. 서 론

2019년 5G시대가 밝았다. 뿐만 아니라 우리의 삶에는 조금씩 빅데이터와 머신러닝로 학습되어진 인공지능 기술이 스며들고 있다. 이러한 IT 기술이 우리의 삶에 스며들기까지 이론적인 부분의 발전과 그에 맞춘 하드웨어적인 발전도 꾸준히 이루어져왔고 이제 아프리카와 서남아시아 등에 여러 대의 대형 드론을 띄어 인터넷 연결이 안되는 지역을 없애겠다는 계획까지 나오고 있다. 엔디비아는 2018년 말 GTX 시리즈가 아닌 실시간 트레이싱, 인공지능, 프로그래밍 가능한 음영을 제공하기 위한 RTX까지 등장하였다. 하지만 기본적으로 인공지능을 위한 하드웨어는 비싸기 때문에 우리는 단일 보드 컴퓨터인 라즈베리 파이(Raspberry Pi)를 통한 좀 더 나은 성능을 위한 방법으로 클러스터를 통한 병렬 처리 방안에 대해 알아보며 그에 대한 문제점과 발전 방향을 제시한다.

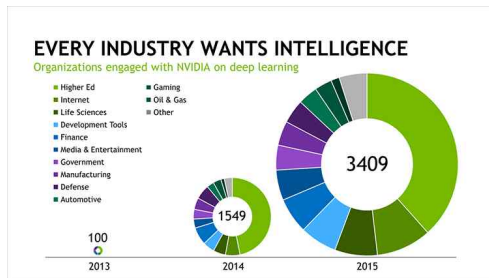


그림 1. 지능적 사업을 위한 기업들의 증가

II. 병렬 처리와 클러스터

인텔의 설립자 중 한명인 고든 무어는 ‘컴퓨터 반도체의 성능은 18개월마다 2배가 된다’라는 무어의 법칙이라는 것을 내놓았다. 이러한 무어의 법칙은 2000년대까지는 맞아 들어가다 점점 작아지는 반도체 속에 많은 트랜지스터를 넣는 것에 비해 비용과 발열 등의 문제가 점점 많아져 프로세서의 속도가 무너지기 시작할 때 하나가 아닌 두 개 이상의 메모리를 공유시키는 방법을 찾았고 그에 맞는 병렬 처리 시스템(Parallel Computing)이 발전하기 시작했다.[1,2]

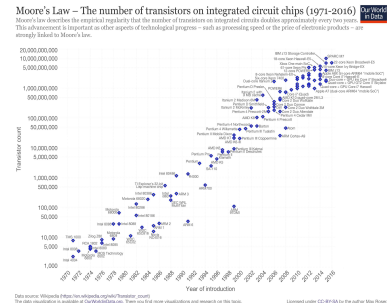


그림 2. 무어의 법칙

이렇게 병렬 처리의 중요성이 대두되며 기업에서도 병렬 처리 기술을 위한 엔디비아의 CUDA같은 기술이 나타나기 시작했다. 구글 또한 늘어나는 데이터와 느려지는 검색 속도 개선을 위하여 더 좋은 성능의 서버가 아닌 많은 노드들이 분산 처리를 하여 결과를 모아 최종 결과를 계산하는 방식인 맵리듀스를 만들었다.

III. 병렬 처리를 위한 기술

우리는 병렬 처리를 위한 기술로서 2가지 방안을 제시한다. MPI, Redis와 Fabric을 이용한 방법이다. MPI(Message Passing Interface)는 분산 및 병렬 처리에서 정보의 교환에 관해 기술하는 표준이다. 메모리를 따로 가지는 프로세스들이 데이터를 공유하기 위해 메시지(데이터)를 송신, 수신하여 통신하는 방식이다. Redis와 Fabric을 이용한 방법의 Redis(Remote dictionary Server)는 “키-값” 구조의 비정형 데이터를 저장하고 관리하기 위한 시스템이다. Redis서버를 통해 우리는 값들을 키에 따라 정리시킴과 Fabric을 통한 원격명령을 통하여 라즈베리파이들이 하나의 서버와 나머지의 클라이언트로서 구성하여 클라이언트들의 자원들을 사용하여 병렬처리를 통해 작업 시간 단축을 할 수 있게 해준다.[3,4]

IV. MPI4PY를 이용한 구현

MPI4PY는 MPI의 통신규약을 Python으로 사용하는 오픈소스이다. 우리는 클러스터링을 위하여 사용할 라즈베리파이들의 Hostname을 pi001, pi002 등으로 일괄적으로 바꾸어주고 MPI 사용을 위해 MPI를 설치해 준다. 그 후 라즈베리파이의 연동을 위해 NMAP 네트워크 작업을 한다. 클러스터링을 할 IP들이 적힌 파일을 만들고 그 후 서로간의 접속을 위한 공개 키(Public Key)와 비밀 키(Private Keys)를 서로간 연동을 해주어 컴퓨터간 접속이 원활하게 해준다. 그리고 명령어인(mpirun -f file -n N python3 file.py)를 실행해주게되면 연결된 라즈베리파이들이 해당 python파일들을 실행한다.[5,6]

IV. Redis와 Fabric을 이용한 구현

Redis는 데이터베이스, 캐시 및 메시지 브로커로 사용되는 오픈소스인 메모리 데이터 구조 저장소이며 동시에 N개의 클라이언트가 수행한 명령 실행을 전체 클라이언트에게 쿼리를 보내 처리하는 병렬 처리 기능이 있다. Redis를 통한 구성은 보다 심플하고 간편하다. 우리는 좀 더 간단한 구현을 위해 (<https://github.com/code-tutorials/rpi-task-queue>)의 자료를 인용하여 구현해보았다. 라즈베리파이의 설정은 ssh 접속 허용을 통한 Fabric의 원격 조정이 가능한 상태를 구성해준다. 우리는 작업시간을 비교할 알고리즘으로 rsa 알고리즘을 생성하고 푸는 작업을 이용할 것이다. 각각의 작업클라이언트에 rsa 서버에는 rsa, RQ, Redis 패키지를 설치한다. 우리는 서버로 사용할 HOST와 클라이언트를 알려줄 수 있는 클라이언트들의 IP들을 지정하여 Queue Task를 주고 병렬처리에 따른 처리속도를 비교해볼 수 있다.[7,8]

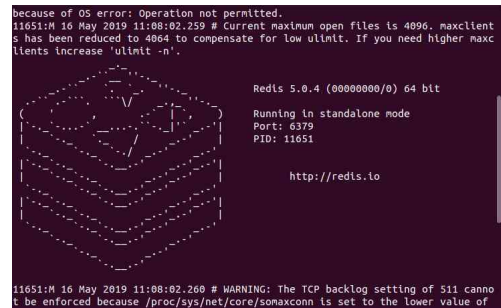


그림 3. Redis 서버 실행

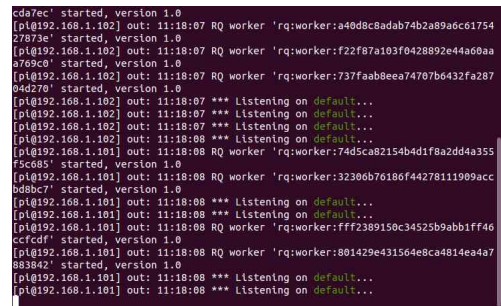


그림 4. 라즈베리파이의 병렬처리 준비

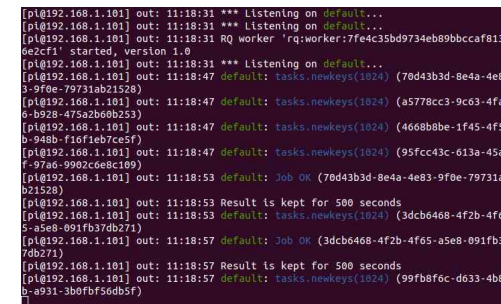


그림 5. 라즈베리파이의 Queue 작업 병렬 처리

우리는 라즈베리파이 1개, 2개, 3개일때의 1024bit rsa 알고리즘 생성 및 처리에 걸리는 시간을 비교하여보았다. 라즈베리파이가 1개일 때 평균 133.9초, 2개일 때 평균 48.0초, 3개일 때 29.7초정도 걸리는 것을 알 수 있었다.

IV. 문제점

우리는 MPI4PY와 Redis, Fabric을 이용한 병렬처리 및 클러스터링을 구현해보았다. MPI4PY를 이용한 구현은 각각의 라즈베리파이간의 통신을 통한 클러스터링을 처리할 수 있다는 것을 알 수 있다.

하지만 병렬 처리를 위해서는 MPI의 Rank가 무엇인지와 해당 Rank에 대한 처리를 일일이 구현시켜야 하는 문제점이 있다. 만약 프로그래밍적인 부분에서 처리를 하지 않을 경우 동시에 같은 작업을 시키는 단점이 있다. Redis와 Fabric을 이용한 구현은 MPI와는 다르게 Redis를 통하여 여러개의 라즈베리파이 임에도 불구하고 하나의 시스템으로 인식하여 전체적으로 심플하고 편리하지만 Redis 서버가 3G 이상의 메모리를 요구하는 단점이 있어 간단한 라즈베리파이로만은 구현할 수가 없다.

V. 결론

우리는 라즈베리파이를 이용한 클러스터링과 병렬처리를 위한 방안에 대해 연구하여 보았다. 우리는 제시한 문제점의 해결방안으로 Redis, Fabric과 MPI4PY를 동시에 이용하여 처리하는 방안을 생각하고 연구하고 있다. 우리가 제시한 문제점을 해결할 경우 라즈베리파이같은 장비를 통하여 우리의 IoT 분야에서 빅데이터 수집 뿐만 직접적인 머신러닝 처리도 같이 이용할 수 있을 것으로 생각한다. 우리는 실생활에서 효율적인 데이터 축적과 인공지능을 이용한 실생활의 만족도 증가, 사물인터넷의 발전 등을 이룰 방안에 대해 고민하고 해결할 것이다.[9]

참 고 문 헌

- [1] 정용재, 김동관. (2014). 라즈베리 파이 클러스터 환경에서의 병렬 프로그래밍. Proceedings of the Korean Society of Computer Information Conference. 477-478.
- [2] 강석원, 김재운, 이상준, 송왕철. (2017). 라즈베리 파이를 이용한 병렬 처리 시스템 구축, 한국콘텐츠학회. 21-22
- [3] 박원돌, 민옥기, 원종호. 무선 센서 네트워크에서 라즈베리 파이를 이용한 분산 협업 시스템, 한국정보과학회. 2015.12. 366-367
- [3] AK Dennis. (2013). RASpberry Pi super cluster.

- [4] MJ Quinn. (2003). Parallel Programing. TMH CSE
- [5] P Pacheco.(1997). Parallel programming with MPI
- [6] WD Gropp, W Gropp, E Lusk, A Skjellum. (1999). Using MPI: portable parallel programming with the message-passing interface
- [7] 오형석. (2017). 자동 API 생성 및 데이터 처리 성능 향상을 위한 RESTful,Redis 기반의 소형 API 플랫폼 설계.
- [8] 허원호, 김은태, 양현대, 박현섭, 정준영. (2015,12). Deep Learning 학습 병렬화 소프트웨어 개발 환경 제공 위한 라즈베리 파이 기반 병렬 컴퓨터 개발. 37-39.
- [9] 공만식, 채홍준, 유보현. (2016). 사물인터넷(IoT) 기술동향과 전망. 기계저널