

수퍼컴퓨터와 Big Data 를 위한 하드웨어의 발전 방향

강갑석 (kskang@ipp.mpg.de)

High Level Support Team (HLST)

Numerical method for Plasma Physics

Max-Planck Institute for Plasma Physics (IPP)

막스플랑크 플라즈마 물리 연구소, 독일

목차

1. 들어가는 말
 2. AI, Big Data 와 수퍼컴퓨터
 3. 하드웨어의 현재와 미래
 - 3.1 필요한 하드웨어의 조건
 - 3.2 현재 사용되는 하드웨어
 - 3.3 프로그래밍 언어, 라이브러리, 그리고 최적화
 - 3.4 앞으로의 하드웨어 발전 방향
 4. 맺는말
- 참고문헌

1. 들어가는 말

사물인터넷 (IoT), 인공지능 (AI), Big Data 는 현재 산업계와 학계를 포함하여 거의 모든 곳에서 이야기 되고 있다. 현재 이 부분들에 관한 이야기는 일상생활에서 편리성을 가져다 줄 것이며 이로 인해 많은 다양한 사업이 가능하며 미래의 먹거리가 될 것이라는 예측과 더불어 일자리를 대체하며 사회를 감시하는 빅브라더가 출현, 그리고 사생활이 침해 될 것이라는 예측으로 정치, 사회, 경제적으로 다양한 이야기들이 나누어 지고 있다.

사물인터넷은 각종 사물에 센서와 통신 기능을 내장하여 인터넷에 연결하는 기술을 의미한다. 사물이란 가전제품, 모바일 장비, 웨어러블 컴퓨터 등 다양한 임베디드 시스템들을 말하며 이 사물들은 자신을 구별할 수 있는 고유한 아이피를 가지고 인터넷에 연결되어야 하고, 데이터 취득을 위한 센서를 내장 할 수 있어야 한다. 이 사물들은 각각이 수집한 많은 정보를 공유하고 분석하여 이를 토대로 주어진 명령을 수행하기도 할 것이다. 이러한 과정에서 모든 사물은 해킹의 대상이 될 수 있어 사물인터넷의 발달과 보안의 발달은 함께 가야 하는 구조이다.

사물인터넷을 이용한 스마트 도시, 스마트 공장, 스마트 사무실등의 개념들이 등장하고 있으며 이는 주어진 상황과 환경에서 자원의 효율성을 극대화 함으로 이루어 질 수 있으며 이는 또한 각각의 기기들이 수집한 Data 를 분석하고 정리하며 각각의 관계성을 인지함으로 실현할 수 있다. 이러한 관계에서 Data 를 분석하여 유용한 방법을 찾아내는 Data Mining 분야의 중요성이 커지고 있다. 특별히 2020년까지 사물의 개수가 260 억개에 이르 것이라는 예상이 있으며 이와 같은 많은 사물이 연결되면 인터넷을 통해 방대한 양의 Data 가 모이게 되는데 이 모인 Data 를 분석하는 Big Data 분석이 필요하게 되었다.

Big Data 와 이를 분석하는 것은 새로운 개념이 아니라 이전부터 의료정보, 생물정보학, 천문망원경으로 부터의 천체 정보, 대단위 실험으로부터의 실험 Data 등등이 존재하였으며 이러한 정보들은 각 분야의 전문가들이 전문지식을 바탕으로 분석하여 왔다. 이러한 분석방법은 제한된 인력만이 분석에 참여할 수 있어 분석할 수 있는 Data 의 양이 한정되며 이를 컴퓨터를 이용하여 할 수 있는 효율적인 알고리즘의 개발이 필요하고 이와 아울러 비정형의 Data 를 포함한 Data 로부터 가치를 추출하고 분석하여 Data 에 가치를 부여하는 기술을 필요로 한다. 이러한 Big data 로 부터 유용한가치를 추출하는 기술로 인공지능(AI)이 유력한 연구 대상으로 지목되고 이를 이용하려는 시도가 많아지고 있다.

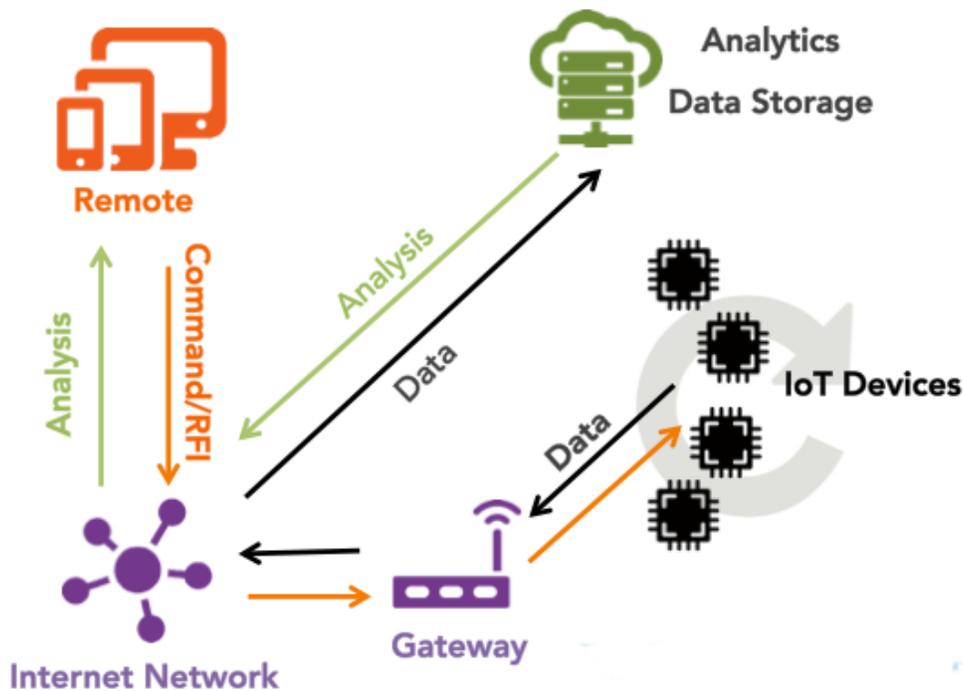
AI 는 기계로부터 만들어진 지능, 즉 시스템이 의해 만들어진 지능을 말하며 때때로 자가학습기능을 가진 시스템을 말한다. 현대 AI 기술은 매우 방대하며 예로는 자율주행자동차, 의학진단, 예술, 수학정리증명, 게임 등등이다. AI 이 자가학습기능을 생각하면 머신러닝을 생각 할 수 있고 이 중 뇌가 뉴런으로 이루어진 전기적인 네트워크라는 생각에 기초하여 나온 신경네트워크에서 발전된 딥러닝 (Deep Learning) 이 활발히 연구되고 있다. 이 딥러닝 방법을 포함한 머신러닝방법의 효용성은 훈련에 사용

되는 Data 에 따라 결정되는 데 이 훈련을 위해서는 방대한 양의 Data 를 필요로 한다. 결국 이는 Big Data 와 연관된다. 또한 훈련을 통한 런닝과정에서 방대한 계산을 필요로 한다. 위와 같이 방대한 Data 를 다루고 방대한 계산을 위해서는 보다 빠른 컴퓨터를 필요로 한다.

이러한 배경을 바탕으로 AI 와 Big Data 관련하여 머신러닝과 슈퍼컴퓨터에 대하여 자세히 알아보고 이에 필요한 하드웨어의 현재와 미래에 대하여 알아보겠다.

2. AI, Big Data 와 슈퍼컴퓨터

앞에서 살펴본 것과 같이 사물 인터넷은 사회적으로, 경제적으로 가능성이 많아 많은 기업, 연구소, 정부를 포함한 공공기관 등에서 시행을 연구 검토하고 있으며 실제로 적용하고 있다. 이러한 사물인터넷을 적용하는 과정을 살펴보면 다음과 같은 과정을 통해 이루어 집니다.



가) Data 수집: 이미지 영상, 온도, 교통상황 등등을 센서를 이용하여 측정하거나 인지하여, 때때로 인지된 자료 중에 필요한 자료만 뽑아내어 인터넷을 통해 Data 를 분석할 곳으로 보내는 과정

나) 분석: 수집된 Data 는 인터넷을 통해 개인 컴퓨터, Data Storage, 서버, 클라우드 컴퓨터등으로 보내지고 이곳에서는 Simulation, Big Data 분석 등등을 통해 상황이 분석 해석되는 과정

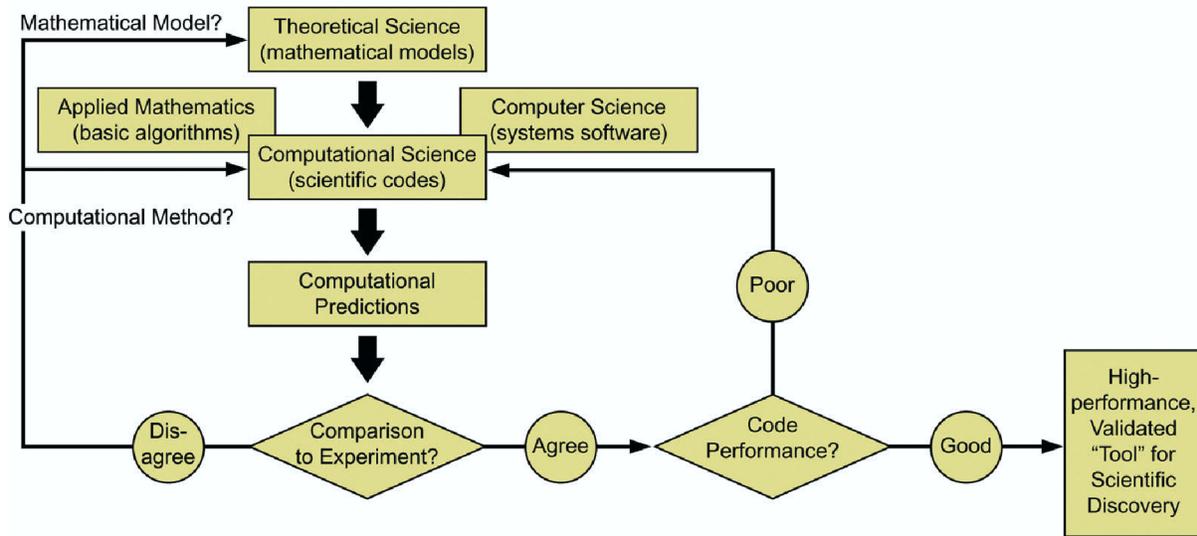
다) 명령: 분석 또는 해석된 상황에 맞추어 전문가시스템이나 AI 등등을 통해 무엇을 해야 하는지 결정하여 명령을 내리는 과정

라) 행동: 인터넷을 통해 전달 된 명령을 수행하는 과정

위의 과정 중 Data 수집은 주로 센서와 관련되고 행동은 수행하는 과정이라 컴퓨터와 직접 관련이 없고 컴퓨터 하드웨어와 관련되어 살펴볼 부분은 분석과 명령입니다. 분석의 과정은 Simulation, Big Data 분석 등의 방법으로 해석되며 이는 개인 컴퓨터나 클라우드컴퓨터, 서버등을 통해 이루어진다.

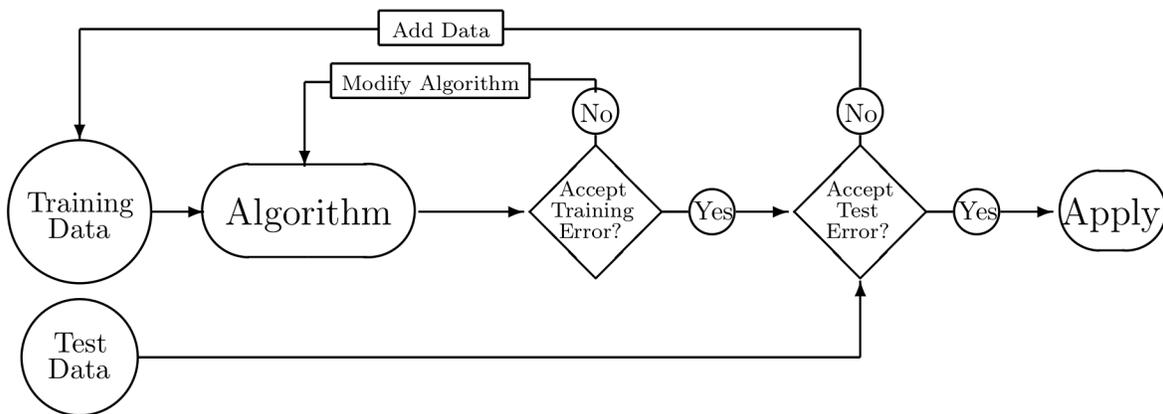
Simulation 방법은 이전부터 계산과학의 중요 수단이며 이는 슈퍼컴퓨터를 이용하여 연구되고 실행되어 왔으며 현재 이 분야는 실험과 관찰, 이론과 더불어 과학의 제 삼의 연구 방법으로 생각되어 지고 있다. 이러한 Simulation 은 Fusion Reactor Simulation, 천체시뮬레이션, 분자 상의 에너지 계산, 기후예측 시뮬레이션 등등 실제 실험을 하기에 불가능하거나, 가격이 많이 들거나, 또는 실험하기에 매우 위험한 연구 분야에서 주로 이루어지고 있다. 이러한 분야들은 매우 많은 계산을 필요로 하며 이를 위해서는 매우 빠른 슈퍼컴퓨터를 필요로 한다.

이 계산과학에서 연구되어지는 과정을 살펴보면 다음과 같은 과정을 통해 이루어진다.

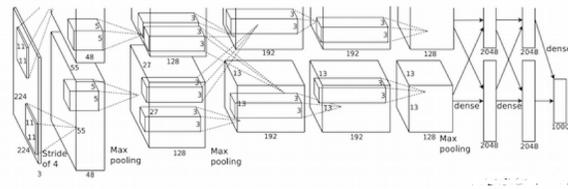
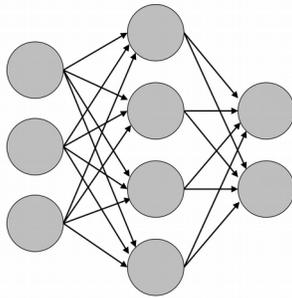


위의 과정을 보면 문제를 수학적 모델로 구현하고 이 수학적 모델을 수치적으로 풀었을 때 그 해가 올바른가 확인해 줄, 문제에 대해 잘 이해하고 있는 응용과학자, 수학적 모델을 풀기 위한 수치적방법을 제공하고 그 수치적 방법이 올바른것인가 확인해 줄 계산수학자, 그리고 이를 컴퓨터상에서 구현하고 최적화 할 컴퓨터과학자를 필요로 한다. 현재 계산과학연구를 하는 데 꼭 필요한 도구인 슈퍼컴퓨터의 구조가 점점 복잡해져서 최적화하는 데에 보다 전문적인 지식과 경험을 필요로 하고 있다.

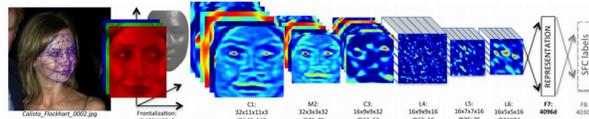
Big Data 분석은 현재 주목받는 분야로 많은 곳에서 투자를 하고 있지만 사실 이 분야는 갑자기 시작된 것이 아니라 오래 전부터, 그리고 여러 분야에서 통계의 방법을 사용하여 연구되어 왔다. 특별히 의료, Genom 프로젝트로 대표되는 생물정보학, 경영, 여론조사 등등에서 Data 의 통계 처리가 보편화 되어 있다. 현재 Genom 으로 대표되는 생물정보학, 천체망원경을 통해 얻은 Data 를 다루는 천체 물리, 핵융합연구의 실험 Data 등등 여러 분야에서는 계속해서 쌓이는 방대한 Data 를 분석하는 것이 난제이며 이를 인력으로 처리하기에는 불가능하다. 이와 같은 방대한 Data 를 처리하기 위해 일찍부터 컴퓨터를 이용한 Data 처리방법이 연구되었다. 이러한 분야의 Data 처리방법은 현재 고전적인 분석방법으로 생각되어지고 있으나 아직도 유용한 방법입니다. 고전적인 분석방법 외에 현재에는 인터넷 쇼핑, 구글링 등으로부터의 방대한 Data 로부터 유용한 연관관계를 알아내어 타겟마케팅에 사용하거나 선거결과의 예측등의 분야에서 Big Data 분석이 필요로 하고 있고 현재 연구되어지고 있다.



이제 Data 분석은 컴퓨터를 이용하여 이루어지는데 고전적인 분석방법 외에 AI 에 의한 Data 분석기법도 사용되고 있다. AI 에 의한 분석기법 중에는 기계적 학습 (머신러닝)을 통하여 분석하는 방법들이 개발되고 있다. 머신러닝 방법으로는 Naive Bayes, SVM, GA, Linear regression, 딥러닝 (Deep Learning) 방법 등등이 있다. 머신러닝이 이루어지는 과정을 보면 주어진 Data 를 이용하여 수학적인 모델을 만드는 학습 (Training) 과정과 학습된 모델을 Data 에 적용하여 결과를 예측하는 적용 (Inference) 과정으로 이루어진다. 현재 가장 많이 관심을 끌고 있고 활발히 연구되어지는 머신러닝 방법은 Deep Learning 방법 (심층학습)이다. 딥러닝은 여러 비선형 변환기법을 통해 높은 수준의 추상화 즉 다량의 Data 나 복잡한 자료들 속에서 핵심적인 내용 또는 기능을 요약하는 작업을 시도하는 Machine learning 알고리즘이다. 어떠한 Data 가 있을 때 이를 컴퓨터가 알아 들을 수 있는 형태로 표현하고 이를 학습에 적용하기 위해 많은 연구가 진행되고 있으며 이 결과로 Deep Neural Networks (DNN), Convolutional Deep Neural Networks (CNN), Deep Belief Networks (DBN) 와 같은 다양한 딥러닝 기법들이 컴퓨터비전, 음성인식, 자연어처리, 음성/신호처리 등등의 분야에 적용되고 있다. 다음 그림은 뉴런망의 변수 개수에 따른 뉴런망의 개념을 그린 도형이다.



60 million parameters



120 million parameters

이 딥러닝의 역사를 살펴보면 1960 년대에 Binary 분석에 뉴런망 (Neural Networks)이 사용되었고 1970 년대에 믿을 만한 결과들을 얻지 못하여 연구가 시들해지다가 1980 년대에 Back Propagation 이 학습방법으로 사용되었다. 2000 년대에는 많은 자료와 더 많은 계산은 연설, 영상이미지 (Vision), 자연어 처리등등 에서 믿을 만 한 결과를 얻어 보다 많은 곳에서 사용되어지고 있다.

그러면 일찍이 시작된 딥러닝이 왜 지금이야 각광을 받게 되었을까? 그이유는 보다 많은 Data 가 축적 되어 학습에 사용될 수 있으며, 보다 빠르고 강력한 하드웨어를 사용할 수 있게 되었고, 보다 발전된 알고리즘이 개발되어 보다 정교한 학습모델이 사용되고 있기 때문이다.

3. 하드웨어의 현재와 미래

3.1 필요한 하드웨어의 조건

앞에서 살펴 보았듯이 딥러닝, 그리고 Big Data 가 주목을 받는 것은 이를 구현할 수 있는 컴퓨터가 발전 하였기 때문이다. 컴퓨터의 발전 상황의 지표로 슈퍼컴퓨터의 발전을 살펴보면 된다. 우선 앞에서 살펴 본 사물 인터넷, Big Data, 계산과학에서 필요로 하는 하드웨어의 필요 조건들을 살펴보자.

사물인터넷의 하드웨어는 사물의 센서와 통신 기능을 기본으로 한다. 센서기능은 목적에 맞는 특별한 기능을 갖고 있어야 하며 경우에 따라서는 유용한 Data 를 선별할 수 있는, 또는 변형하는 기능을 추가로 필요로 한다. 이러한 기기는 일반적으로 소형이고 쉽게 기능을 향상 변형할 수 있는 조건을 필요로 한다.

Big Data 와 머신러닝은 상대적으로 간단한 계산으로 방대한 Data 를 다루어야 한다. 방대한 Data 를 다루려면 이에 따른 방대한 메모리를 필요로 하고 이 메모리를 빠르게 다루는 것을 필요로 한다. 반면

계산은 단순하여 Co-processor 나 Accelerator 등 작고 단순한 Core 를 사용하여도 된다. 이러한 계산은 때때로 클라우드컴퓨터를 통하여 이루어질 수 있다.

계산과학에서의 Simulation 은 일반적으로 복잡한 계산을 필요로 하며 방대한 양의 계산을 필요로 한다. 이 방대한 양의 계산을 수행하기 위해서는 많은 컴퓨터 노드와 Core 를 병렬로 연결하여 수행해야 하며 이 과정에서 노드간에 Data 를 주고 받아야 한다. 따라서 계산을 빠르게 할 수 있는 Core 와 Data 를 빠르게 주고 받을 수 있는 Data Communication 장치를 필요로 하며 이 Data Communication 은 노드 수가 증가함에 따라 너무 급격하게 증가하지 않아야 한다 (Scalability). 하지만 노드 수가 증가함에 따라 Data Communication 에 필요로 하는 시간이 늘어나는 것은 피할 수 없으므로 이 Data Communication 을 피할 수 있는 알고리즘도 연구되고 있다.

3.2 현재 사용되는 하드웨어

현재 사용되는 하드웨어를 살펴보기 위해 현 서버와 슈퍼컴퓨터의 구성요소와 구조를 살펴보자. 우선 컴퓨터는 입출력 (IO)기기, 저장매체, 그리고 명령수행과 계산을 위한 Core 로 구성되는 CPU 가 있다. 이러한 여러 개의 CPU 는 Node 를 구성하고 이 Node 들을 Network 으로 연결하여 하나의 컴퓨터를 구성하는 Cluster 형태가 주를 이루고 있다. 지금부터 Cluster 각각의 구성 요소별로 현재 사용되는 하드웨어를 살펴보자.

먼저 컴퓨터에서 가장 중요한 계산을 담당하는 Core 를 살펴보면 다음과 같이 구분 할 수 있다.

가) 중앙처리장치 (CPU): 중요한 제어장치로 계산과 명령을 내리고 입출력을 제어하는 장치를 말한다. 현재 Intel 에서는 Braodwell, Skylake Microarchitecture 를 기반으로 한 Xeon Brand 가 있고, AMD 에서는 Zen microarchitecture 를 기반으로 한 Ryzen Brand 가 있다. 지금의 CPU 들은 8-40 개의 Core 를 가지고 있다.

나) Co-Processor 또는 Accelerator: 계산기능을 주로 하는 Core 로 많은 양의 계산을 비교적 간단한 Processor 에서 나누어서 하는 Core 를 말하며 현재 NVIDIA 의 Graphical Processoing Unit (GPU) 가 있다. NVIDIA 의 Tesla K20X 는 2688 개의 Core 로 구성되어 있고, 약 1.31 Tflops/s 의 계산 성능을 가지고 있으며 이를 지원하기 위한 CUDA, OpenCL 언어들이 있다. 그 외 PetaPath, Cell Processor 등의 Accelerator 가 있다.

다) Light Weight Core: 계산 기능을 주로하지만 간단한 명령수행도 가능한 Core 를 말하며 대표적으로 ARM 기반 CPU 와 Intel Many Integrated Core (MIC) 인 Xeon Phi, IBM 의 BlueGene Core 등이 해당된다. IBM 의 BlueGene Core 는 일찍이 슈퍼컴퓨터용의 제작되었으며, ARM 기반 Core 는 저전력 Core 로 전화기에 주로 사용되어 왔고 이를 기반으로 한 슈퍼컴퓨터를 제작하는 프로젝트가 유럽

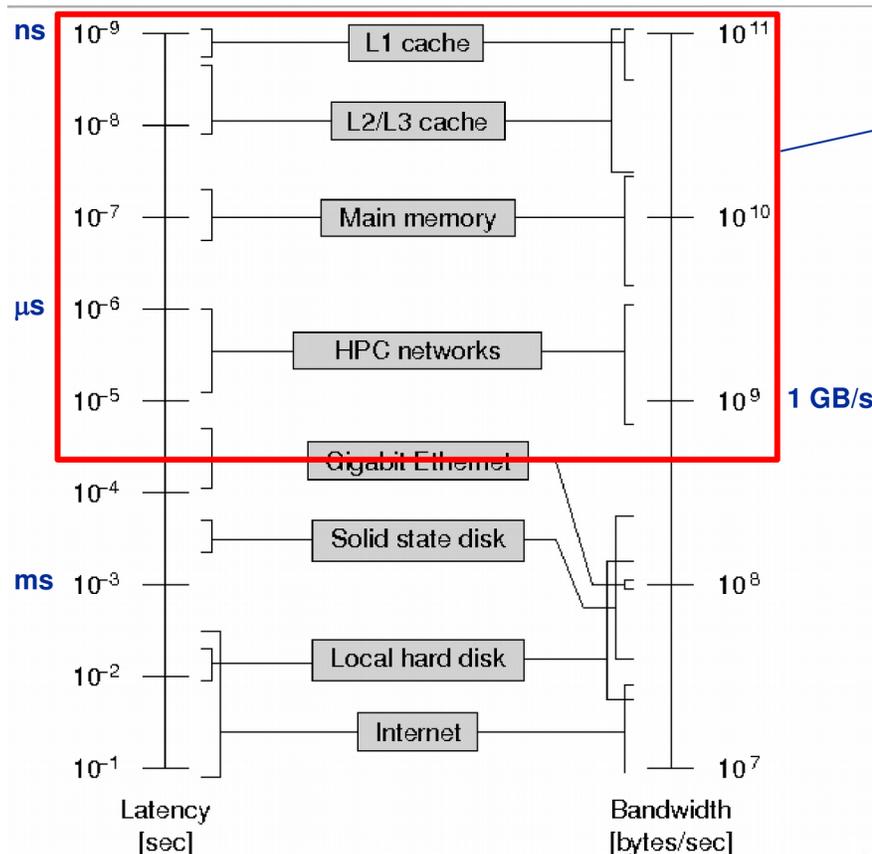


에서 바르셀로나 슈퍼컴퓨터센터에서 Mont-Blanc 프로젝트로 진행되었고 최근에 영국 Bristol 대학에 Cray 사에서 Isambard 라는 기계 설치를 시작했다. Intel 의 Xeon Phi 는 Knight Core (KNC)와 Knight Landing (KNL)이 제작 판매되고 있다.

라) ASIC (Application-specific integrated Circuit) chip: 특별한 기능에 맞게 제작된 Chip 을 말하며 여러 회사에서 소용량으로 제작되고 있으며 대표적으로는 대용량의 계산을 작은 정확도로 빠르게 계산할 수 있는 Google 의 Tensor Processing Unit (TPU)이다.

마) FPGA (Field-Programmable Gate Array): 공장에서 만들어진 후에 설치 된 곳에서 고객이나 디자이너가 특정한 목적에 맞추어 바꿀 수 있는 Chip 을 말한다. 이로한 변형성 때문에 이 FPGA Chip 은 다양한 분야에서 사용되어왔다. 현재 이 Chip 은 훈련을 통한 최적화된 알고리즘을 쉽게 구현할 수 있어 그 유용성이 증가하고 있다.

다음으로 컴퓨터에는 Data 를 저장하는 메모리가 중요하며 이 메모리는 L1 Cache, L2/L3 Cache , 그리고 주메모리로 구성된다. 이 각 단계의 메모리들은 메모리 저장 용량과 Data Communication 속도에서 차이가 난다. 또한 Node 간의 Data Communication, 하드디스크에 Data 쓰기과 읽기, 그리고 인터넷을 통한 Data Communication 도 중요하다. Data Communication 에서 성능은 Communication 을 시작하기 위한 명령을 수행하는 Latency 와 최대로 전달할 수 있는 Data 양을 나타내는 Bandwidth (BW) 가 중요하다. 일반적으로 Latency 와 BW 는 반비례한다.



위의 Core 들과 메모리의 조합으로 Node 가 구성되는데 주로 CPU 만으로 또는 이질적인 하드웨어를 조합한 CPU + (GPU, MIC, ARM, FPGA) 등으로 구성될 수 있다. 또한 이 Node 들은 HPC Network 으로 연결되어 서버나 슈퍼컴퓨터를 구성한다. 각각의 경우의 Latency 와 BW 를 정리하면 다음의 그림 에서 나타내는 것과 같다. 일반적으로 BW 가 크고 Latency 가 작을 수록 가격이 비싸서 실제 적용에는 제한적이다.

현재 슈퍼컴퓨터의 Node 들을 연결하는 HPC Network 을 슈퍼컴퓨터 Top500 리스트에서 보면 Infiniband, Cray 사의 Aries 와 Gemini, Intel 의 Omni Path 등이 많이 사용되었으며, 일본의 Fujitsu 사의 K-Computer 에 사용된 Tofu, 현재 가장 빠른 컴퓨터인 중국의 Sunway TaihuLight 에 사용된 Sunway, Tianhe-2 에 사용된 TH Express-2, 그리고 IBM BlueGene 에 사용된 연결 방법 등등이 있다. 기술개발의 성과를 보면 BW 는 많이 향상되었으나 Latency 를 줄이는 기술은 상대적으로 덜 발달되었고 향상 속도가 더디다.

3.3 프로그래밍 언어, 라이브러리, 그리고 최적화

하드웨어와 더불어 알고리즘을 수행하게 하는 소프트웨어도 중요하다. 이제 현재 많이 사용되는 프로그래밍 언어와 라이브러리, 그리고 이의 최적화에 대해 살펴보자.

기본적으로 프로그래밍언어는 수치계산에 주로 사용하는 Fortran, 그리고 C 와 C++ 등등 하이레벨 언어들 이 사용되고 있다. 이 하이레벨 언어들 을 바탕으로 하드웨어에 적합하고 많이 사용되는 특별한 기능들을 모아 놓은 라이브러리들이 사용되고 있다. 또한 코딩 된 프로그램은 최적화가 되어야 한다. 특별히 프로그램의 최적화는 하드웨어를 따라야 하므로 하드웨어에 대한 이해가 필요하고 따라서 하드웨어를 만든 회사가 최적의 라이브러리를 제공하기도 한다. 현재 Intel 에서는 Intel Compiler 와 기본적인 Math Library 인 MKL 을 제공하고 있고 GPU 를 만드는 NVIDIA 는 GPU 에서 프로그래밍을 위한 프로그래밍 언어인 CUDA 라이브러리를 제공하고 있다. 프로그램을 여러 다양한 그리고 이질적인 컴퓨터에서 수행하기 위해서는 표준화가 필요하고 이 표준화를 따르면 쉽게 여러 컴퓨터에서 수행 할 수 있다.

프로그램 최적화는 주어진 컴퓨터사양에서 자원을 최대한으로 사용할 수 있게 하며 이는 Core, Node, Cluster 단계별로 나누어서 생각 할 수 있다. 최적화하는 과정은 우선 어느 단계 또는 함수가 시간을 많이 사용하는가 알아내는 프로파일링 과정, 그리고 서로 다른 Compiler 최적화 옵션을 이용하는 과정, 병렬화와 이에 맞는 Scalable 한 알고리즘과 새로운 프로그램 구조의 사용등을 생각할 수 있다.

우선 Core 단계에서 살펴보면 Register 와 Execution Unit 사용의 최적화, 그리고 Cache Miss 를 최소화 하는 방법과 주어진 최적화 옵션에서 높은 단계를 사용하는 방법이 있다. 또한 같은 연산을 반복적으로 해야 하는 경우 한번에 연산을 수행하는 벡터연산법 (Vectorization) 또는 SIMD (Single

Instruction Multiple Data)이 있으며 이 방법을 위해서는 Data 정렬이 필요하지만 정규적 격자구조에 보다 효과적이다. 한 Core 위에서 여러 Thread 를 연속적으로 사용하여 명령단계에서 주어진 자원을 최대한으로 이용하는 Simultaneous Multithreading 방법이 사용되고 있다.

Node 단계의 최적화로는 Shared Memory 시스템에서 사용되어 온 OpenMP 라이브러리가 있다. OpenMP 는 같은 메모리를 공유하는 Core 들에서 주 Thread 가 프로그램을 수행하다가 병렬계산을 할 수 있는 부분에서 여러 Thread 를 실행하여 함께 연산을 수행하는 방법으로 1997 년부터 개발되어져 오고 있다. OpenMP 라이브러리는 Fortran, C, C++언어에서 사용되며 현재 버전 4.0 의 표준화가 진행되고 있다.

Cluster 단계 최적화로는 Data 분산과 Communication 을 다루는 MPI (Message Passing Interface) 라이브러리가 주로 사용되고 있다. MPI 라이브러리는 1994 년부터 개발되어 사용되고 있으며 계속하여 발전하며 표준화되고 있다. 표준화를 통하여 한 시스템에서 개발된 프로그램을 다른 시스템에서도 쉽게 사용할 수 있다. 이 단계에서는 Load Balance 와 Data Communication 을 줄이는 알고리즘 개발을 필요로 한다. 실제 시스템에서 특히 5000 개이상의 MPI Task 를 가지는 경우 Data Communication 을 위한 시간이 오래 걸리며 이를 줄이기 위해서 하이브리드 MPI/OpenMP 방법이 사용되고 있다.

3.4 앞으로의 하드웨어 발전 방향

현재 개발되고 있는 하드웨어를 살펴보면 우선 CPU 의 Clock 속도를 늘리는 것은 기술적으로 한계에 도달하여 연산량은 MPI, OpenMP, Vectorization 을 통한 분산연산을 통해 증가시키고 있다. 또한 Clock 속도에 따른 필요한 에너지량은 Clock 속도의 제곱에 비례하여 증가하고 이용가능한 에너지의 한계로 인해 필요한 에너지를 줄이기 위해 Clock 속도를 줄이고 Core 수를 늘리는 방법으로 연산량을 증가시키고 있다. 이를 위해 한 Core 안에 좀 더 많은 계산 Unit, 기본 함수계산 Unit 과 Vector Unit 을 늘리는 방향으로 설계가 되고 있다.

각 Node 당 더 많은 메모리가 사용되고 있지만 Core 수가 늘어남에 따라 각 Core 가 사용할 수 있는 메모리 양은 줄어들고 있다. 메모리 사용을 위한 Latency 줄이는 기술과 보다 큰 BW 를 사용하는 기술이 개발되고 있고 이를 위해 메모리를 Chip 위에 설치하는 Memory on Chip 기술이 사용되고 있다. 또한 하드웨어의 일시정지를 줄이기 위한 기술, 부분적으로 실패한 하드웨어를 감지하고 다른 부분으로 대체해 계산하는 기술들이 개발되고 있다.

수퍼컴퓨터는 각각의 분야에서 필요한 Simulation 알고리즘을 주어진 하드웨어에 맞추어 최적화하는 연구가 진행되고 또한 그 Simulation 에 따라 현재 주어진 기술로 최적화된 그리고 가격을 최소화 할 수 있는 하드웨어를 설계하는 Co-Design 개념으로 연구가 진행되고 있다. 이 개념은 Big Data, 그리고

인공지능을 위한 하드웨어에도 적용되고 있다. 즉, 좀 더 많은 메모리와 Data 를 사용 할 수 있도록 하고 간단한 연산을 분산처리로 할 수 있는 하드웨어들이 설계되고 있으며 이에 최적화된 라이브러리와 알고리즘들이 개발되고 있다.

NVIDIA 에서는 이미 GPGPU 를 이용한 슈퍼컴퓨터와 Big Data 를 위한 하드웨어와 이를 최적화 할 수 있는 CUDA 언어와 이를 위한 라이브러리가 제공되어 있어 많은 곳에서 사용되고 있으며 이를 계속해서 최적화하고 있다. Intel 에서는 슈퍼컴퓨터를 위한 Xeon Processor 와 Xeon Phi Processor 와 이에 최적화 된 Compiler 와 OpenMP, MPI, MKL 라이브러리들이 제공되고 있으며 Big Data 를 위한 Nervana 가 개발되고 있다. 이 Nervana 는 좀 더 많은 메모리와 이를 최적화하여 사용할 수 있는 하드웨어로 개발되고 있고 하드웨어와 함께 최적화 된 MKL 을 포함한 Machine Learning Scaling Library (MLSL)도 함께 개발되고 있으며 하드웨어와 함께 제공하고 있다.

4. 맺는말

앞에서 Big Data 를 위한 하드웨어와 슈퍼컴퓨터의 발전 방향에 살펴보았다. 하드웨어는 점점 복잡해지고 특별히 특정 목적을 위해 가격을 포함하여 최적화하여 구성하여야 하며 이를 위해서는 각 하드웨어 부분들이 어떠한 강점이 있고 또한 기술적으로 어느 선까지 발전 할 수 있는가 알 수 있는 전기전자분야의 전문가, 각 하드웨어에 맞추어 프로그램을 최적화 할 수 있는 전산과학자, 문제에 알맞는 알고리즘을 개발할 수 있는 수치해석학자, 문제의 목적과 나온 결과가 올바른가 확인할 수 있는 응용과학자들의 협력을 필요로 한다.

하드웨어는 각 부분들의 성능도 중요하지만 가격 또한 중요하다. 특히 가격과 관련해서는 대부분 반도체 관련 사업의 특징인 많은 개발비와 대량 생산 공정으로 시장에서 얼마나 많은 부분을 점유하느냐가 가격을 결정하는 중요한 요소이다. 이를 위해 몇몇 회사에서는 응용분야를 넓히는 노력도 하고 있다. 예로는 NVIDIA 에서 GPU 를 슈퍼컴퓨터의 Accelerator 로, 또 머신러닝에 이용할 수 있도록 프로그램 언어인 CUDA 와 라이브러리를 개발했 오고 있다. 또한, 저전력 설계로 주로 전화기에 사용되어 대용량으로 생산되어 가격을 낮추어 온 ARM 기반 CPU 를 기반으로 한 슈퍼컴퓨터가 개발되고 있다. 이 또한 머신러닝에 적합한 하드웨어를 개발하기 위한 프로젝트도 진행되고 있다.

참고문헌

1. 위키백과 ko.wikipedia.org
2. www.intel.com
3. www.nvidia.com
4. www.top500.org
5. The Internet of Things: An Overview. ([www. Internetsociety.org](http://www.Internetsociety.org))



Wednesday, 16 August 2017

6. A Full hardware Guide to Deep Learning. (news.zyoyene.com)
7. V. Sze, Y.H. Chen, J. Emer, A. Suleiman, and Z. Zhang, Hardware for Machine Learning: Challenges and Opportunities, arXiv:1612.07625v3
8. A machine learning application landscape and appropriate hardware alternatives, Moor Insights & Strategy
9. D. Keyes, Algorithmic Adaptations to Extreme Scale.
10. D. Keyes, Applications that break techniques, Sir James Lighthill Distinguished Lectureship in Mathematical Sciences

