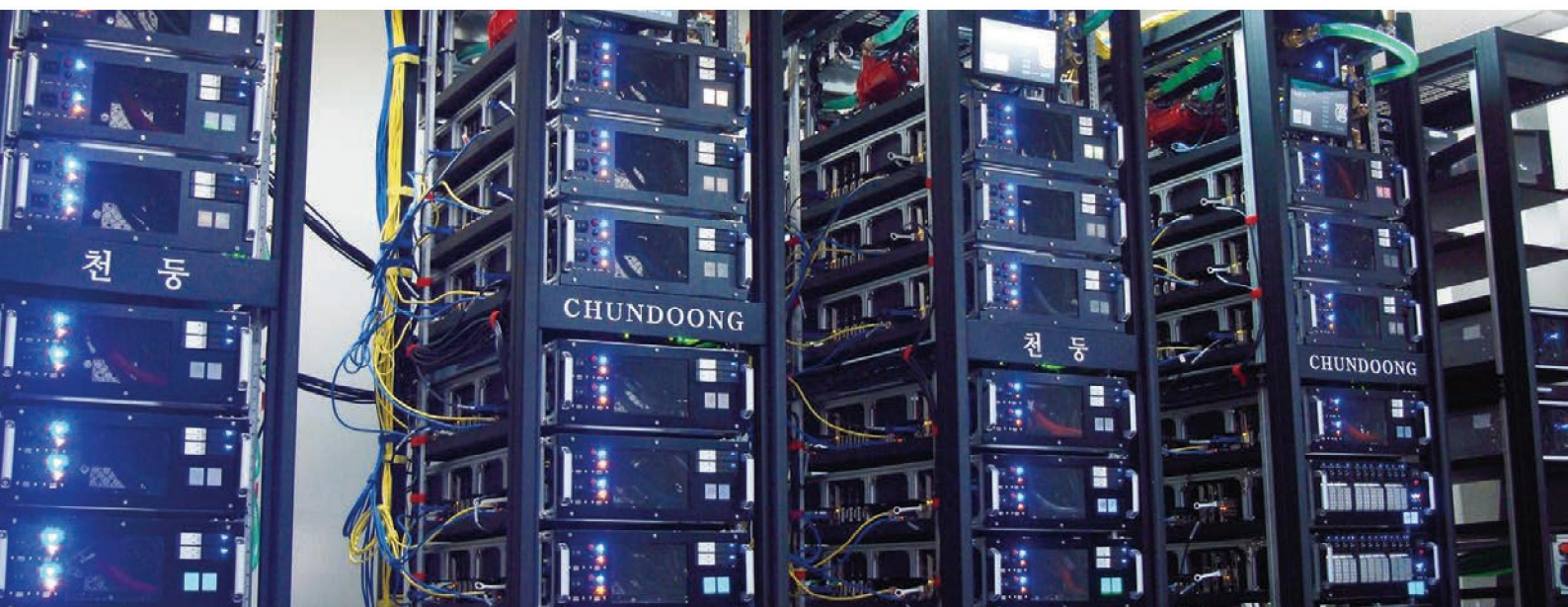


국내 슈퍼컴퓨팅 산업 2

이종 슈퍼컴퓨터 기술 동향과 슈퍼컴퓨터 천둥의 개발 및 운영 사례

조강원 (서울대학교 컴퓨터공학부), 이재진 교수 (서울대학교 컴퓨터공학부)



1. 슈퍼컴퓨터 기술 동향

슈퍼컴퓨터란 당대에 세계에서 가장 계산 성능이 높은 컴퓨터 및 이와 비슷한 수준의 컴퓨터들을 일컫는다. 여기서 '비슷한'이라는 말에 모호함이 있는데, 보통은 Top500 리스트[1]에 오른 컴퓨터들을 슈퍼컴퓨터라고 부른다. Top500은 전 세계 컴퓨터들을 린팩 벤치마크(LINPACK benchmark)로 측정한 FLOPS(floating point operations per second, 초당 부동소수점 연산 횟수) 단위의 계산 성능을 기준으로 1위에서 500위까지 순위를 매기는 리스트로, 매년 6월과 11월 두 차례 발표된다. 즉, 세계에서 500위 내에 드는 계산 성능을 가지고 있으면 슈퍼컴퓨터로 분류한다고 이해할 수 있겠다.

슈퍼컴퓨터는 보통의 컴퓨터로 풀기 어려운 대규모의 계산을 가능하게 하여 과학기술 연구, 신제품 개발 등의 분야에 널리 사용되며, 더 빠른 슈퍼컴퓨터를 보유함으로써 해당 국가, 기관 혹은 기업의 경쟁력을 높일 수 있다. 따라서 미국, 일본, 중국, EU 등의 선진국들을 중심으로 슈퍼컴퓨터 성능 경쟁이 치열하게 이루어지고 있다. 그림 1은 Top500 리스트가 처음 발표된 1993년부터 지금까지 슈퍼컴퓨터들의 성능이 어떻게 변화해 왔는지를 나타내는데, 매년 약 1.9배의 성능 향상이 이루어졌음을 확인할 수 있다. 이는 무어(Moore)의 법칙에 의해 프로세서의 성능이 높아지는 속도(18개월에 2배)를 훨씬 상회하는 수치로, 슈퍼컴퓨터 기술이 얼마나 빠른 속도로 발전해 왔는지를 잘 나타내 준다.

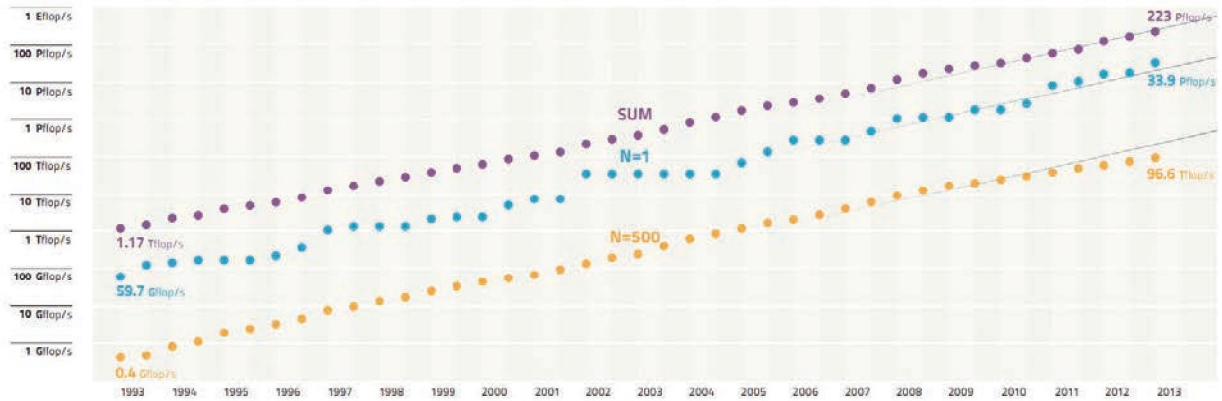


그림 1. Top500 슈퍼컴퓨터의 성능 추이 [1]

슈퍼컴퓨터의 성능이 높아짐에 따라 그 시스템 구조 역시 많은 변화를 겪어 왔다. 1980년대까지 슈퍼컴퓨터는 주로 소수의 벡터 프로세서가 메모리를 공유하고 있는 SMP(symmetric multiprocessing) 구조였다. 하지만 이는 시스템 규모를 확장하는 데 한계가 있어, 1990년대부터는 수백 개 이상의 프로세서가 독립적인 메모리를 가지며 인터커넥션 네트워크로 연결되어 있는 MPP(massively parallel processing) 구조가 널리 사용되었다. 2000년대 들어서는 슈퍼컴퓨터의 확장성(scalability)과 가용성(availability)을 높이고 구축비용 및 유지보수비용을 줄이는 것이 중요한 문제로 등장하였다. 이에 따라 여러 대의 독립적인 컴퓨터를 인터커넥션 네트워크로 연결하고 소프트웨어적으로 통합해 단일 시스템처럼 동작하게 하는 클러스터 구조의 사용이 보편화되었다. 또한 하드웨어 기술의 발전으로 고성능 계산에 특화된 전용 하드웨어가 아닌 일용품(commodity) 하드웨어를 사용해 슈퍼컴퓨터를 구축할 수 있게 되었다. 일례로 2013년 6월 기준 Top500 슈퍼컴퓨터 중 452대가 Intel과 AMD의 x86 프로세서를 사용하며, 423대가 Ethernet, InfiniBand 등의 상용 인터커넥션 네트워크를 사용한다.

이렇듯 과거에는 슈퍼컴퓨터가 주로 성능과 규모를 키우는 방향으로 발전해 왔다면, 최근에는 슈퍼컴퓨터의 전력효율을 높이는 것이 중요한 이슈로 대두되고 있다. 현재 세계에서 가장 빠른 슈퍼컴퓨터인 중국 Tianhe-2의 소비전력은 17.8 MW로 이미 웬만한 소도시 하나의 소비전력과 맞먹는 수준이다. 과거의 슈퍼컴퓨터 성능 추이로 미루어 봤을 때 2020년 전후로 엑사스케일 슈퍼컴퓨터, 즉 1초에 실수 연산을 10^{18} 번 수행 가능한 컴퓨터가 등장할 것이라 예상된다. 하지만 현재의 기술 수준이 그대로 유지된다면 이 시스템의 소비전력은 수 GW 수준으로, 원자력발전소 하나를 새로 지어야만 슈퍼컴퓨터를 운영할 수 있는 꼴이 된다. 미국 국방부의 슈퍼컴퓨터 개발 프로젝트를 총괄하는 DARPA(Defence Advanced Research Projects Agency)는 현실적인 기반시설 확보 가능성을 감안했을 때 엑사스케일 슈퍼컴퓨터의 소비전력 목표를 20 MW로 제시한 바 있다[2]. 이를 달성하기 위해서는 지금보다 슈퍼컴퓨터의 전력효율을 100배 수준으로 향상시켜야 한다.

2. 이종 슈퍼컴퓨터의 등장

슈퍼컴퓨터의 전력효율을 높이기 위한 방안 중 하나로 서로 다른 종류의 프로세서를 함께 사용하는 이종(heterogeneous) 시스템이 주목을 받고 있다. 기존 슈퍼컴퓨터가 CPU만으로 이루어져 있던 것과 달리 이종 슈퍼컴퓨

터는 CPU와 가속기(accelerator)를 함께 장착한다. 여기서 가속기란 특정 유형의 계산을 저전력으로 매우 빠르게 수행하는 데 특화된 프로세서를 가리킨다. CPU는 운영체제를 실행하고 I/O를 담당하며 대용량의 계산은 가속기에서 수행함으로써 슈퍼컴퓨터의 성능과 전력효율을 높일 수 있다. 이중 슈퍼컴퓨터는 2006년 6월 Top500 리스트에 처음으로 등장하였으며, 그림 2와 같이 그 수가 늘어나는 추세이다. 2013년 6월 Top500 리스트에는 총 54대(10.8%)의 이중 슈퍼컴퓨터가 등재되어 있으며, 특히 세계 1위인 중국 Tianhe-2와 2위인 미국 Titan을 포함 상위 10대 슈퍼컴퓨터 중 40%가 이중 시스템으로, 미래에는 이중 슈퍼컴퓨터의 사용이 더욱 늘어날 것임을 시사한다.

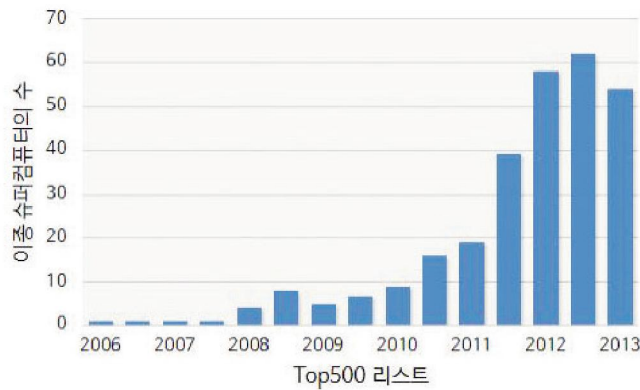


그림 2. Top500 리스트에 등재된 이중 슈퍼컴퓨터 수

이중 슈퍼컴퓨터에서 사용되는 가속기의 가장 대표적인 예로는 GPU를 꼽을 수 있다. GPU는 2013년 6월 Top500에 등재된 이중 슈퍼컴퓨터 중 약 80%에서 가속기로 사용된다. GPU는 원래 그래픽 처리 전용의 프로세서였으나, GPU의 구조가 프로그래밍 가능하도록 변화하면서 GPU를 범용 계산에 사용하는 GPGPU(General-Purpose computing on Graphics Processing Units) 기술이 등장하였다. 일반적으로 GPU에는 수백~수천 개의 간단한 스칼라(scalar) 코어가 내장되어 있다. 코어 하나의 성능은 얼마 되지 않지만, 모든 코어를 동시에 사용하면 많은 양의 계산을 병렬로 한꺼번에 수행할 수 있다. GPGPU 기술이 주목을 받으면서 NVIDIA Tesla와 같이 아예 그래픽 출력 없이 계산 용도로만 사용할 수 있는 GPU도 등장하였으며 오늘날 이중 슈퍼컴퓨터에는 주로 이러한 계산 전용 GPU가 사용된다.

이중 슈퍼컴퓨터에서 많이 사용되는 또 다른 가속기로는 2012년 말에 출시된 Intel Xeon Phi 코프로세서를 들 수 있다. Xeon Phi 코프로세서에는 약 20년 전에 펜티엄 프로세서에서 사용되던 P54C 코어 60개 내외가 집적되어 있다. 단, GPU와 비슷하게 많은 양의 계산을 동시에 수행할 수 있도록, 각 코어마다 4개의 하드웨어 스레드를 실행할 수 있는 기능과 512비트 데이터에 대해 벡터 연산을 수행하는 기능이 추가되었다. 기존에 CPU를 대상으로 작성한 프로그램을 큰 수정 없이 컴파일, 실행할 수 있는 것이 장점이다.

표 1은 최신 CPU와 GPU, Xeon Phi 코프로세서를 하나씩 골라 비교한 것이다. 가속기를 이루는 각 코어는 CPU보다 낮은 클럭에서 동작하고 구조도 단순하여, 코어 하나의 이론 최대 성능(Rpeak, 모든 계산자원을 100% 사용했을 때의 성능)은 CPU보다 뒤떨어진다. 하지만 코어 수가 CPU보다 많아, 결과적으로 전체 프로세서의 이론 최대 성능은 CPU보다 훨씬 높다. 실제 계산 성능이 반드시 이론 최대 성능과 일치하지 않음을 감안하더라도, 병렬도가 높

은 코드는 가속기에서 더 빠르게 실행될 것임을 알 수 있다. 더불어 가속기의 성능이 CPU보다 5~7배 정도 높은 데 반해 소비전력은 2배를 넘지 않는데, 이는 같은 양의 계산을 가속기에서 CPU보다 낮은 전력으로 수행할 수 있음을 시사한다.

	CPU	GPU	Xeon Phi
제품명	Intel Xeon E5-2690	NVIDIA Tesla K20x	Intel Xeon Phi 5110P
코어 클럭	2.90GHz	732 MHz	1,053 GHz
코어 수	8	2688	60
코어 당 이론 최대 성능	23.2 GFLOPS	0.49 GFLOPS	16.85 GFLOPS
전체 이론 최대 성능	185.6 GFLOPS	1310 GFLOPS	1011 GFLOPS
최대 전력	135 W	235 W	225 W

표 1. 최신 CPU, GPU, Xeon Phi 코프로세서 비교

Top500에 오른 슈퍼컴퓨터들의 전력효율 순위인 Green500 리스트[3]에서 이종 슈퍼컴퓨터가 가지는 장점을 확인할 수 있다. 슈퍼컴퓨터의 전력효율은 소비전력 대비 계산성능(FLOPS/W)으로 정의되며, 이는 동일한 에너지를 사용해 얼마나 많은 계산을 수행할 수 있는지를 알려 준다. 2013년 6월 기준으로 Green500 리스트의 1~4위는 모두 이종 슈퍼컴퓨터이며, 그 중 두 대는 세계 최초로 3 GFLOPS/W대의 전력효율을 달성하였다. 또한 Green500의 상위 50대 시스템 중 21대(42%)가 이종 슈퍼컴퓨터이다. 전체 Top500 리스트에서 이종 슈퍼컴퓨터가 차지하는 비율이 10.8%임을 감안할 때, 이종 슈퍼컴퓨터의 전력효율이 대체로 높은 편임을 알 수 있다.

3. 슈퍼컴퓨터 천동의 개발 사례

천동은 이종 시스템을 위한 소프트웨어 기술을 테스트하기 위한 목적으로 서울대학교 매니코어 프로그래밍 연구단에서 2012년에 개발한 이종 슈퍼컴퓨터이다(그림 3). 56대의 노드로 이루어진 클러스터로, 각 노드에는 CPU 2개와 GPU 4개가 장착되어 있다. 천동은 린팩 벤치마크에서 106.8 TFLOPS(1012 FLOPS)의 성능을 얻어 2012년 11월 Top500 리스트의 277위에 올랐다. 또한 1,870 MFLOPS/W의 전력효율로 Green500에서 역대 국내 최고 기록인 32위를 달성하였다. 특히 국내의 다른 슈퍼컴퓨터들이 대부분 외국 업체에서 도입한 것인 데 비해 천동은 시스템 설계·구축의 전 과정이 독자적으로 이루어졌다는 점에 의의가 있다. 보통 슈퍼컴퓨터를 도입하는 데는 수십~수백억 원이 넘는 비용이 들고 전기비만 해도 1년에 수천만 원이 넘어간다. 하지만 천동의 경우에는 7억 원 가량의 제한된 연구비로 시스템을 구축하고 운영비용도 최소화해야 했다. 따라서 기존 슈퍼컴퓨터보다 가격효율(가격 대비 성능)과 전력효율(소비전력 대비 성능)을 더욱 높일 수 있는 독자적인 모델 개발이 필요했다.



그림 3. 슈퍼컴퓨터 천동의 모습

시스템의 효율성을 높이기 위해, 천둥은 크게 두 가지 측면에서 기존 이중 슈퍼컴퓨터와의 차별화를 꾀했다. 첫째로, 전체 시스템에서 가속기(GPU)가 차지하는 비중을 극대화하였다. 가속기는 일반적으로 CPU보다 가격효율 및 전력효율이 뛰어나다. 따라서 이중 슈퍼컴퓨터에 CPU 대비 가속기를 많이 장착할수록 구축비용과 소비전력을 줄일 수 있다. 하지만 암달의 법칙(Amdahl's Law)에 의해 가속기의 비중을 계속 늘린다고 해서 애플리케이션 성능을 무한정 빠르게 만들 수는 없다. 표 2에서 알 수 있듯이 기존의 이중 슈퍼컴퓨터에서 가속기와 CPU의 성능 비는 많아야 12:1을 넘지 않는데, 격차가 더 벌어질 경우 CPU가 주된 성능 병목이 되어 가속기의 성능을 잘 발휘하지 못하기 때문이다. 천둥에서는 GPU가 차지하는 비중을 더 늘려 효율성을 높이고, 대신 CPU 부분의 수행시간을 단축하는 소프트웨어 최적화 기법을 개발·적용하는 전략을 택하였다. 개발한 최적화 기법을 슈퍼컴퓨터의 성능 측정에 사용되는 린팩 벤치마크에 적용한 결과, GPU와 CPU의 성능 비가 약 15:1에 이르더라도 CPU에 의한 병목이 거의 없이 고성능을 얻을 수 있음을 확인하였다. 따라서 각 노드에 GPU를 4대씩 장착하고, CPU로는 Intel Sandy Bridge-EP 8코어 프로세서 중 가장 성능이 낮고 비용이 저렴한 Intel Xeon E5-2650을 사용하기로 결정하였다.

이중 슈퍼컴퓨터	노드 내 CPU 구성	노드 내 가속기 구성	CPU Rpeak (TFLOPS)	가속기 Rpeak (TFLOPS)	가속기 / CPU
중국 Tianhe-2	2 × Intel Xeon E5-2692	3 × Intel Xeon Phi 31S1P	6758	48144	7.12
미국 Titan	AMD Opteron 6274	NVIDIA Tesla K20x	2631	24481	9.30
미국 Stampede	2 × Intel Xeon E5-2680	Intel Xeon Phi SE10P	2212	7386	3.34
일본 TSUBAME 2.0	2 × Intel Xeon X5670	3 × NVIDIA Tesla M2050	198.0	2176	10.99
사우디 SANAM	2 × Intel Xeon E5-2650	2 × AMD FirePro S10000	76.80	887.0	11.55
천둥	2 × Intel Xeon E5-2650	4 × AMD Radeon HD 7970	14.34	212.1	14.80

표 2. 이중 슈퍼컴퓨터에서 CPU와 가속기의 이론 최대 성능(Rpeak) 비교

둘째로, NVIDIA Tesla와 같은 고가의 계산 전용 GPU 대신 일반적인 그래픽 출력용 GPU를 사용하여 구축비용을 크게 절감하였다. 일반 GPU도 하이엔드급은 OpenCL이나 CUDA와 같은 프로그래밍 모델을 지원하므로 계산 전용 GPU와 똑같은 방법으로 범용 계산에 사용할 수 있다. 성능 측면에서도 둘 사이에 큰 차이가 없다. 그럼에도 불구하고 이중 슈퍼컴퓨터들이 대부분 계산 전용 GPU를 사용했던 것은, 메모리 용량이 크고 ECC(error-correcting code) 메모리를 지원하는 등의 장점이 있기 때문이다. 천둥에서는 계산 전용 GPU보다 가격이 1/10 이상 저렴한 AMD Radeon HD 7970 GPU를 사용하고, 대신 일반 GPU가 가지는 한계를 극복하기 위한 방안을 모색하였다. 예를 들어 GPU 메모리 용량이 작아 한 번에 저장·처리할 수 있는 데이터의 양에 한계가 있으므로, GPU가 수행할 계산을 여러 부분으로 쪼개고 커널 실행과 PCI-E 통신을 중첩하여(overlapping) 통신으로 인한 지연을 없애는 소프트웨어 최적화 기술을 적용하였다. 후술할 냉각 시스템을 사용해 GPU의 온도를 낮게 유지함으로써 메모리에서의 에러 발생 확률을 대폭 줄였다. 또한 고성능 서버용 마더보드와 게이머들이 주로 사용하는 하이엔드 GPU 사이에 호환성 테스트가 충분히 이루어지지 않았을 공산이 크므로, 사전에 노드 4개로 이루어진 소규모 프로토타입을 구축하여 직접 하드웨어 호환 여부를 확인하고 문제 발견 시 마더보드 제조사와의 협력을 통해 해결하였다.

이상의 두 가지 전략을 바탕으로 계산 성능과 효율성을 최대화하는 이중 슈퍼컴퓨터를 설계·구축하였다. 하드웨어 구성요소를 선정하는 데는 린팩 벤치마크의 성능을 기준으로 삼았는데, 그 이유는 (1) 천둥은 프로그래밍 모

델, 런타임 시스템, 컴파일러 등 소프트웨어 기술의 테스트를 목적으로 하므로 특정 응용 분야의 애플리케이션을 대상으로 삼기가 부적절하고 (2) 린팩 벤치마크가 Top500을 비롯 슈퍼컴퓨터의 성능 측정에 널리 사용되며 (3) 선형대수 문제를 푸는 프로그램이므로 과학기술 애플리케이션의 평균적인 추세에 부합할 것이라 판단했기 때문이다. 그 결과 각 노드에는 1,600 MHz로 동작하는 128 GB의 메인 메모리를 장착하고, CPU와 GPU 사이는 PCI-E 3.0 버스로 연결하며, 노드 간 연결을 위해서는 InfiniBand QDR 스위치 6개로 이루어진 fat tree 네트워크를 구성하였다. 한편 다른 슈퍼컴퓨터의 계산 성능 대비 스토리지 용량 비율을 참고하여 88TB 규모의 Lustre 파일 시스템을 제공하는 스토리지 노드 4대를 추가로 설치하였다.

슈퍼컴퓨터의 개발에 있어 중요한 문제 중 하나는, 좁은 공간에 밀집된 다수의 프로세서에서 발생하는 열을 효율적으로 방출하는 것이다. 특히 천동에 사용된 AMD Radeon HD 7970 GPU의 경우, 본래 그래픽 출력 용도로 개발된 제품이었으므로 노드 안에 여러 GPU가 함께 장착된 상황에서는 방열이 잘 이루어지지 않았다. 이로 인해 시스템을 오래 가동할수록 GPU의 온도가 올라가 성능이 점점 떨어지고, 최종적으로는 작동이 멈춰 버리는 문제가 발생하였다. 이를 해결하기 위해 물을 이용한 이중 슈퍼컴퓨터용 냉각 시스템을 자체 개발하여 천동에 적용하였다. 실외의 냉각기에서 차갑게 식힌 물을 CPU와 GPU에 접촉한 방열판으로 흘려보내 온도를 내리고, 데워진 물은 다시 실외로 순환시킨다. 급속 분리 밸브를 설치해 냉각 시스템의 일부를 쉽게 분리·교체하고, 온도 센서와 임베디드 PC를 이용해 냉각이 잘 이루어지는지 모니터링할 수 있도록 하였다. 그 결과 CPU는 40°C 이하, GPU는 50°C 이하로 안정적으로 유지할 수 있게 되었으며, 시스템의 안정성을 높이고 부품의 기대 수명을 연장하는 효과를 얻었다. 이러한 방식의 냉각 시스템은 일본 K Computer와 같이 CPU만 사용하는 슈퍼컴퓨터에 적용된 적은 있으나, 이중 슈퍼컴퓨터에서는 노드 내부 구조가 복잡하여 거의 시도된 적이 없었다. 최소한의 공간을 차지하면서 노드 내부의 CPU와 GPU 방열판을 연결하고, 클러스터의 모든 노드로 원활하게 물이 순환하도록 하며, 누수와 결로를 방지하기 위해 1년이 넘는 연구와 시행착오가 이루어졌다.

천동의 개발 사례는 국내의 기술과 경험으로도 세계 중위권의 슈퍼컴퓨터를 자체개발할 수 있음을 증명해 준다. 나아가 단순히 외국의 사례를 답습하는 것이 아니라, 차별화된 설계와 소프트웨어 기술로 우수성을 확보할 수 있음을 시사한다. 천동은 2012년 11월 Top500 리스트 발표 당시 '세계에서 7번째로 전력효율이 높은 슈퍼컴퓨터 구조'로 소개되었다[4]. 천동을 이루는 노드 한 대의 계산 성능은 1.907 TFLOPS로, 2012년 11월 Top500에 등재된 클러스터 규모의 슈퍼컴퓨터 412대 중에서 가장 높다. 이는 천동의 설계를 이용하면 같은 수의 노드를 사용해서 더 높은 성능을 얻을 수 있음을 뜻한다. 무엇보다 천동은 구축비용의 측면에서 다른 슈퍼컴퓨터보다 우위에 있다. 천동의 규모를 키우면 현재 국내에서 가장 빠른 기상청 해온·해담과 동급의 슈퍼컴퓨터는 약 20억 원에, 세계 30위 내

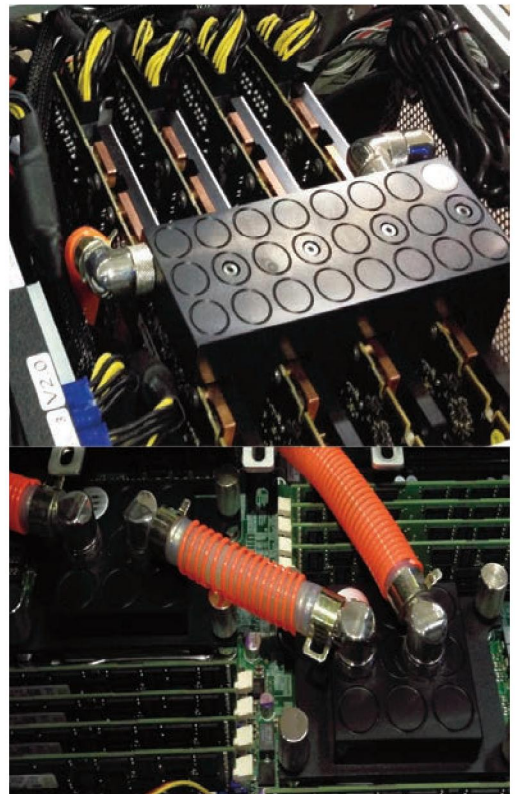


그림 4. 천동의 GPU(위)와 CPU(아래)에 방열판이 부착된 모습

에 드는 1 PFLOPS(1015 FLOPS) 성능의 슈퍼컴퓨터는 약 70억 원에 구축할 수 있을 것으로 보인다. 다만 이보다 성능이 좋은 슈퍼컴퓨터를 개발하려면 천둥에서 맞닥뜨리지 못한 새로운 종류의 문제가 발생하고 이를 해결하기 위한 추가 연구가 필요할 것으로 전망된다.

4. 천둥의 운영 및 활용

천둥은 약 5개월간의 내부 시범운영을 거쳐, 2013년 4월부터 일반인도 연구교육 목적으로 사용할 수 있도록 공개되었다. 이는 국내에서 자체개발한 슈퍼컴퓨터로 일반인에게 서비스를 제공하는 첫 사례이다. 천둥 서비스 제공을 위해 AMD GPU를 지원하는 작업 스케줄러, 시스템 모니터링 소프트웨어, 웹 인터페이스 등을 자체 개발하였으며 피드백을 받아 지속적으로 개선해 나가고 있다. 이들 프로그램의 이름은 북유럽 신화에 나오는 천둥의 신 이름을 따서 토르(Thor)라고 지었다. 7개월이 넘는 기간 동안 큰 문제없이 서비스가 지속되었으며, 2013년 10월 말 기준으로 60명 이상의 사용자들이 천둥에서 염기서열 분석, 유체 시뮬레이션 등의 작업을 수행하고 있다. 한편 서울대학교 매니코어 프로그래밍 연구단에서 주최한 ‘2013 Accelerator Programming 겨울학교’와 ‘2013 국가슈퍼컴퓨팅 Summer School @ SNU’, 중소기업청 취업연계 교육과정인 ‘국산 슈퍼컴 천둥을 활용한 병렬 프로그래밍 전문가 과정’ 등 여러 교육 프로그램을 위해 천둥을 프로그래밍 실습 환경으로 제공함으로써, 이종 컴퓨팅 분야의 전문 인력을 양성하는 데 기여하고 있다. 천둥의 사용 방법과 사용료 등에 대한 자세한 설명은 홈페이지 (<http://chundoong.snu.ac.kr/>)에서 확인할 수 있다.

Job ID	Task ID	Task Name	Command	Status	Start Time	End Time	Completion Time	Exit Code	Nodes	GPU
45001	742 (sc42)	task_45001	thor --add --mode single --device gpu --base-dir here ./mm	Finished	2013-10-31 15:46:20	2013-10-31 15:46:21	2013-10-31 15:46:31	Not deleted	c44	GPU
45000	740 (sc40)	test3	thor --add --mode mpi --base-dir here --name test3 --nodes 4 ./test3	Finished	2013-10-31 15:46:03	2013-10-31 15:46:06	2013-10-31 15:46:14	Not deleted	c40,c41,c42,c43	CPU
44999	742 (sc42)	task_44999	thor --add --mode single --device gpu --base-dir here ./mm	Finished	2013-10-31 15:43:26	2013-10-31 15:43:29	2013-10-31 15:43:39	Not deleted	c40	GPU
44998	504 (jungwon)	task_44998	thor --add --mode mpi --device gpu --nodes 4 --base-dir here ./mm	Finished	2013-10-31 15:43:03	2013-10-31 15:43:17	2013-10-31 15:43:53	Not deleted	c18,c19,c20,c21	GPU
44997	504 (jungwon)	task_44997	thor --add --mode mpi --device gpu --nodes 4 --base-dir here ./mm	Finished	2013-10-31 15:43:02	2013-10-31 15:43:11	2013-10-31 15:43:49	Not deleted	c14,c15,c16,c17	GPU
44996	504 (jungwon)	task_44996	thor --add --mode mpi --device gpu --nodes 4 --base-dir here ./mm	Finished	2013-10-31 15:43:02	2013-10-31 15:43:08	2013-10-31 15:43:41	Not deleted	c06,c09,c12,c13	GPU
44995	504 (jungwon)	task_44995	thor --add --mode mpi --device gpu --nodes 4 --base-dir here ./mm	Finished	2013-10-31 15:43:00	2013-10-31 15:43:04	2013-10-31 15:43:38	Not deleted	c04,c05,c06,c07	GPU
44994	742 (sc42)	task_44994	thor --add --mode single --device gpu --base-dir here ./mm	Finished	2013-10-31 15:42:45	2013-10-31 15:42:49	2013-10-31 15:42:59	Not deleted	c44	GPU
44993	504 (jungwon)	task_44993	thor --add --mode mpi --device gpu --base-dir here --nodes 4 ./mm	Finished	2013-10-31 15:42:17	2013-10-31 15:42:30	2013-10-31 15:42:36	Not deleted	c14,c15,c16,c17	GPU
44992	504 (jungwon)	task_44992	thor --add --mode mpi --device gpu --base-dir here --nodes 4 ./mm	Finished	2013-10-31 15:42:16	2013-10-31 15:42:24	2013-10-31 15:42:29	Not deleted	c06,c09,c12,c13	GPU
44991	504 (jungwon)	task_44991	thor --add --mode mpi --device gpu --base-dir here --nodes 4 ./mm	Finished	2013-10-31 15:42:14	2013-10-31 15:42:20	2013-10-31 15:42:24	Not deleted	c04,c05,c06,c07	GPU
44990	740 (sc40)	test3	thor --add --mode mpi --device gpu --nodes 4 --name test3 --base-dir here ./test3	Finished	2013-10-31 15:42:02	2013-10-31 15:42:07	2013-10-31 15:42:40	Not deleted	c40,c41,c42,c43	GPU
44989	740 (sc40)	test3	thor --add --mode mpi --device gpu --nodes 4 --name test3 --base-dir here ./test3	Finished	2013-10-31 15:40:51	2013-10-31 15:40:55	2013-10-31 15:41:29	Not deleted	c41,c42,c43,c44	GPU

그림 5. 토르(thor)를 사용해 천둥에서 실행 중인 작업을 웹으로 확인하는 모습

5. 결론

지금까지 슈퍼컴퓨터 기술 동향과 최근 이종 슈퍼컴퓨터가 주목받게 된 배경에 대해 살펴보았다. 또한 국내에서 자체개발한 이종 슈퍼컴퓨터인 천둥의 개발 및 운영 사례를 소개하였다. 특히 천둥이 어떤 점에서 기존 슈퍼컴퓨터와의 차별성을 가지는지를 논하였다. 천둥의 설계를 뒷받침했던 각종 소프트웨어 최적화 기술들은 서울대학

교 매니코어 프로그래밍 연구단에서 개발한 공개 소프트웨어인 SnuCL[5, 6]에 통합하여 공개할 예정이다.

슈퍼컴퓨터 기술은 시간이 지남에 따라 서버 및 데스크탑 컴퓨터로, 나아가 모바일 기기로 이전되어 IT 분야 전체의 발전을 이끌고 있다. 또한 슈퍼컴퓨터는 국가 안보 및 재난 대처와 밀접한 연관을 가지며, 다른 과학기술 및 산업 분야에서 대규모 계산을 가능하도록 하는 국가 경쟁력의 핵심 요소이다. 과거 우리나라는 외국 기업에서 슈퍼컴퓨터를 도입, 활용하는 데에만 집중하고 슈퍼컴퓨터 기술의 연구개발을 등한시했던 것이 현실이다. 하지만 앞서 언급하였듯이 연구개발의 진입장벽이 높은 프로세서, 메모리, 마더보드, 인터커넥션 네트워크 등의 하드웨어 부품은 일용품(commodity)화 된 지 오래이며, 이들을 통합하여 효율적인 시스템을 구성하는 설계 및 소프트웨어 기술이 슈퍼컴퓨터 개발의 핵심이 되었다. 특히 이종 슈퍼컴퓨터를 위한 소프트웨어 연구는 전 세계적으로 막 시작하는 단계에 있다. 천둥의 사례에서 확인할 수 있듯이 지금부터 연구개발을 시작하더라도 충분히 세계와 경쟁이 가능하다.

Cray사를 창업하였으며 슈퍼컴퓨터의 아버지라 불리는 Seymour Cray가 한 말을 소개하며 마무리하고자 한다. "Anyone can build a fast CPU. The trick is to build a fast system."

■ Reference

- [1] TOP500 Supercomputer Sites, <http://www.top500.org/>.
- [2] P. Kogge, K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, K. Hill, J. Hiller, S. Karp, S. Keckler, D. Klein, R. Lucas, M. Richards, A. Scarpelli, S. Scott, A. Snavely, T. Sterling, R. S. Williams, and K. Yelick. "Exascale Computing Study: Technology Challenges in Achieving Exascale Systems," Exascale Computing Study Report, 2008.
- [3] The Green500 List, <http://www.green500.org/>.
- [4] "Highlights of the 40th TOP500 List," Presentation of the 40th TOP500 List, <http://www.slideshare.net/top500/presentation-of-the-40th-top500-list>.
- [5] J. Kim, S. Seo, J. Lee, J. Nah, G. Jo, and J. Lee. "SnuCL: an OpenCL Framework for Heterogeneous CPU/GPU Clusters," Proceedings of the 26th International Conference on Supercomputing, pp. 341-352, 2012.
- [6] SnucL: An OpenCL Framework for Heterogeneous Clusters, <http://snucL.snu.ac.kr/>.