기술기사

구조역학 분야에서의 초병렬 컴퓨터의 활용

조 진 우*

최 성 훈*

1. 초병렬 처리

수퍼컴퓨터의 역사는 1972년 Seymour Cray 박사에 의해 설립된 Cray Research, Inc.(CRI)로부터 시작되었다. CRI는 1976년 CRAY-1을 발표, 세계최초로 수퍼컴퓨터를 탄생시킨 이후 1994년 T90 시리즈를 발표하기까지 수퍼컴퓨터의 대명사가 되었다. Cray와 ETA등 미국에 의해 주도되어 오던 수퍼컴퓨터 시장은 1982년부터 하드웨어 제조 기술이 발달한 일본의 Fujitsu, Hitachi, NEC 등의 참여로 치열한 경쟁체제에 돌입하게 되었다. 수퍼컴퓨터 공급 현황을 보면 1990년 통계로 CRI가 세계시장의 67%를 점유하고 있고, 일본의 Fujitsu가 17%를 보유, CRI와 일본 3개사의 경쟁으로 압축된다. 일본의 3개사는 주로 일본시장을 중심으로 시장기반을 확충해가고 있으나 응용 소프트웨어의 부족으로 인해 전세계적으로는 CRI사 제품에 밀리고 있다.

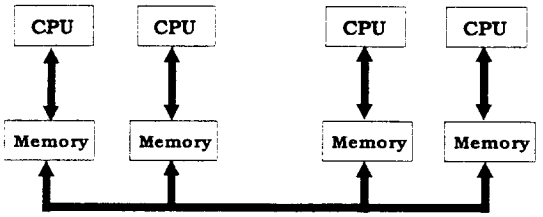
이러한 기존의 벡터처리(Vector Processing : VP) 형태의 수퍼컴퓨터에 일대 혁신을 가져온 것이 초병렬처리(Massively Parallel Processing : MPP) 컴퓨터이다. 병렬처리라 함은 컴퓨터의 성능을 높이기 위해 여러 개의 프로세서를 사용하는 연산을 말하는 것으로, 계산과정에서 발생하는 여러 가지 일 중 병렬로 처리될 수 있는 부분들을 동시에 처리하는 parallelism과 이를 순차적으로 처리하는 pipelining으로 분류할 수 있다.[1] 벡터형 수퍼컴퓨터도 여러 개의 프로세서를 동시에 사용하기도 하지만 여러 프로세서들 간의 상호 결합 기술과 병렬 프로그래밍 기술을 토대로 수십 개에서 수천개 까지의 프로세서를 채용한 컴퓨터를 초병렬 수퍼컴퓨터로 분류한다.

현재 많이 쓰이는 대부분의 MPP는 MIMD (Multiple Instruction Multiple Data)구조를 가지고 있는데 이는 다시 프로세서와 메모리의 관계에 따라 분산 메모리 (distributed memory)구조와 공유 메모리(shared memory) 구조로 세분된다. 분산 메모리 구조는 그림 1(a)에 보인 바와 같

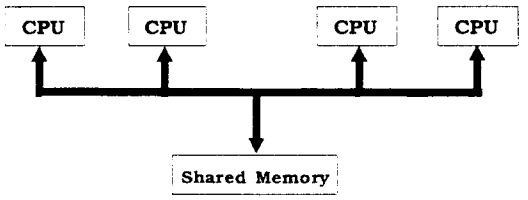
* 정회원, 삼성종합기술원 수퍼컴응용실

이 각각의 프로세서가 자신만의 메모리를 가지고 이 프로세서-메모리 쌍이 네트워크에 의해 상호 연결되어 있는 구조로서, 다른 프로세서에 속한 remote memory를 사용하기 위해서는 사용자에 의해 message passing을 하여야 한다. 이 구조를 갖는 MIMD 컴퓨터는 네트워크에 여러 개의 프로세서-메모리 쌍을 추가로 연결할 수 있어 성능향상을 용이하게 할 수 있는 장점이 있으나 프로그래밍을 위해서는 구조 등에 대한 기술적인 지식이 요구된다. 이와 같은 구조를 가지는 병렬 컴퓨터로는 IBM SP2나 Intel Paragon 등이 있다. 그림 1(b)는 여러개의 프로세서가 네트워크를 통하여 하나의 메모리 시스템에 연결된 공유 메모리 구조를 보인다. 이러한 공유 메모리 구조를 가지는 컴퓨터는 분산 메모리 구조를 가지는 컴퓨터에 비해 프로그래밍이 용이하며 explicit message passing이 필요 없는 장점이 있으나 여러 개의 프로세서가 하나의 메모리를 사용할 경우 병목현상이 생길 수 있으며 프로세서를 추가 할 수 없다는 약점이 있다. Cray C90, Cray T3D 등이 여기에 속한다.

MPP는 10년 전에 Intel사에서 첫 모델을 발표하였고, 지금은 IBM, Cray, Intel, Convex 등이 주로 생산하고 있다. 1994년 자료[2]에 따르면



(a) 분산 메모리 시스템



(b) 공유 메모리 시스템

그림 1 MPP의 프로세서와 메모리 구조

MPP 보급률이 미국에서는 전체 수퍼컴퓨터의 51%, 유럽에서는 32%, 일본에서는 13%이며 계속 증가하는 추세이다. 반면 한국에서는 고작 2~3대가 보급 되어있어 MPP 응용 기술에 대한 많은 투자가 요구되는 실정이다. 미국에서의 MPP 활용의 대표적인 사례는 3대 자동차 회사에서 5개 국립 연구소와 컨소시움을 형성하여 MPP 이용 기술을 연구하고 있는 것으로서, 유체역학, 구조역학, grid 생성기술, 가시화 등의 과제에 적용되고 있다.

계산 속도 및 기억 용량에서 거의 한계에 다다른 VP 기종에 비해 MPP 기종은 발전의 여지가 많으며 다음과 같은 장점을 가진다. 첫째, 최고 성능 면에서 10내지 20GFlops가 최대인 VP 기종에 비해 MPP 기종은 300 내지 400GFlops의 처리 속도를 가질 수 있다. 둘째로 MPP기종은 컴퓨터를 대체하지 않고 프로세서 모듈만을 추가 함으로써 성능을 수백배까지 향상시킬 수 있는 장점이 있다. 결과적으로 같은 성능의 경우 MPP 기종의 가격이 VP 기종 가격의 10분의 1 정도로 저렴하다. 이러한 장점에도 불구하고 MPP가 쉽게 쓰이지 않는 이유는 병렬처리 알고리즘을 적용한 소프트웨어의 부족에 원인이 있다. 우선, 효율적인 병렬처리 프로그램을 작성하는데 어려움이 있을 뿐만 아니라 MPP 기종사이에 호환성이 부족하여 다른 모델의 기종에서 사용 할 수 없는 문제가 있다. 이런 연유로 현재는 MPP에서 사용가능한 상용 소프트웨어가 많지 않으나, 앞서 언급한 장점들로 인해 차세대 수퍼컴퓨터의 주종을 MPP가 차지할 것이라는 전망 하에 많은 소프트웨어 업체들이 기존의 상용 소프트웨어에 대한 병렬화 작업에 착수하고 있다.

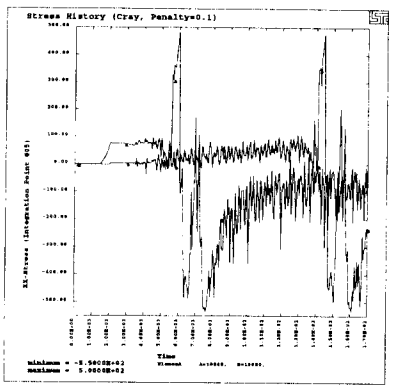
2. 초병렬 컴퓨터의 활용 사례

삼성종합기술원은 91년에 도입한 VP 수퍼컴퓨터인 Cray Y-MP의 업청난 load를 분산시키고자 256개 CPU를 가지고 있는 MPP 수퍼컴퓨터인 Intel Paragon을 95년에 새롭게 도입하여 구조, 열/유체, 진동/음향분야의 시뮬레이션에 활용하

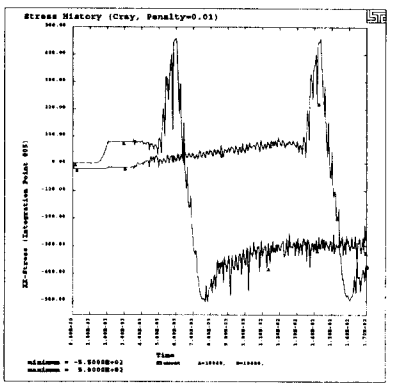
고 있다.[3] 현재 세계적으로는 초병렬 계산에 대한 연구가 활발히 이루어지고 있으며, 그 일환으로 기존의 응용 소프트웨어를 병렬화 시킨 새로운 version들이 각 업체의 개발로 속속 등장하게 되었고, 현재 Paragon에는 Cray에서 사용되고 있는 응용 소프트웨어 중 몇 가지가 설치되어 있는 상태이다. 그러나 Cray에 익숙해져 있는 기존의 사용자들은 paragon의 사용을 기피하고 있으며 병렬화 된 응용 소프트웨어의 신뢰성에도 약간의 문제가 있어서 paragon의 도입취지와는 달리 그 활용도는 아직도 낮은 상태에 머물러 있다.

LSTC사에서 개발한 LS-DYNA3D도 병렬화 된 후 paragon에 설치되었는데, LS-DYNA30는 Cray에서 가장 많은 사용시간을 점유하는 소프트웨어 중 하나이기 때문에 충분히 활용할 경우 Cray의 load를 크게 줄일 수 있는 소프트웨어이다. 그러나 도입 초기 실시한 benchmark의 결과, Cray version과 비교해 볼 때 해에서 많은 차이를 보였으며 몇 가지의 오류가 발견되어 지금까지 사용을 거의 할 수 없는 상태였다. 그러나 LSTC사를 방문하여 지금까지 발견한 오류들을 직접 알리는 한편, 약 6개월 동안 꾸준한 접촉을 거치면서 이번에 새로운 version이 입수되었고, 검증 결과 해의 타당성이 인정되어 실질적인 사용이 가능하게 되었다.

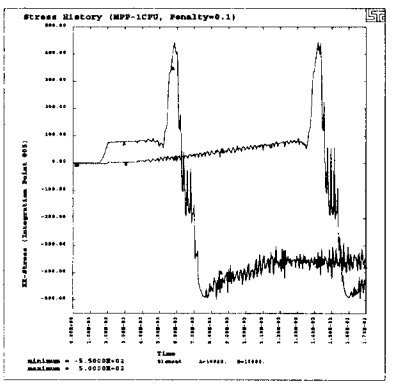
그러므로 본 글에서는, 앞서 실시한 benchmark의 결과를 통해, 병렬화된 LS-DYNA3D의 효율성과 해의 검증을 보여주므로 해서 구조해석 분야에서의 초병렬 컴퓨터의 활용을 높이려는데 그 취지가 있다.



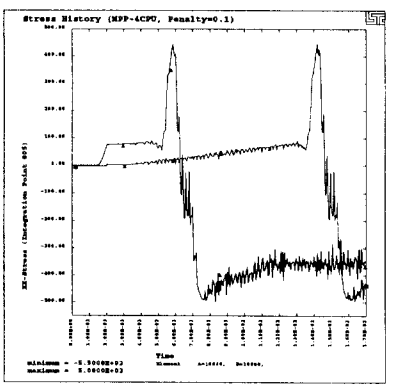
(a) Cray, Penalty=0.1



(b) Cray, Penalty=0.01



(c) MPP-1CPU, Penalty=0.1



(d) MPP-4CPU, Penalty=0.1

그림 2 Stress history의 비교

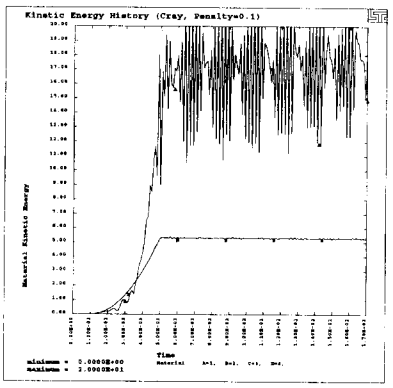
2.1 해의 검증

검증을 위해 선정한 문제는 박판성형의 일종인 2D draw bending문제이다. 이것은 NUMISHEET '93 학회의 Benchmark test[4]로서 Mattiasson[5]이 NUMIFORM '95학회에 제출한 논문에 자세히 기술되어 있기 때문에 여기서는 이론적인 측면보다는 Cray version과의 비교에 역점을 두어 설명하고자 한다.

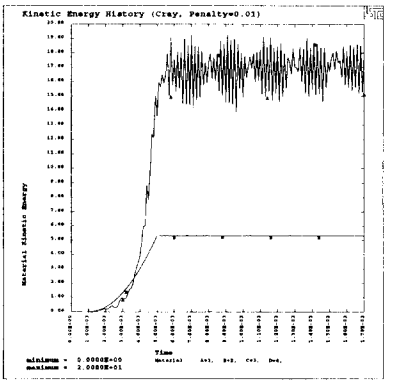
그림 2의 (a)와 (b)는 Cray로 계산한 결과로서 박판 특정부분에서의 시간에 따른 stress를 보여주고 있다. 접촉을 수반한 해석에서의 한 물체가 다른 물체를 침투하지 못하도록 penalty 함수를 사용하게 되는데 두 그림의 차이는 penalty parameter의 값을 바꾼 것에 기인한 것이다. 한편 (c)와 (d)는 paragon으로 계산한 결과로서 두 경우 모두 penalty parameter의 값은 (a)의 경우와 같고 다만 CPU의 사용개수가 틀리다. Mattiasson[5]의 논문에 의하면 (a)보다는 (b)의 결과가 정확한 것이며 (c)를 (b)와 비교해 보면 상당히 유사하다. 이것으로 미루어 볼 때 paragon에서의 계산결과는 타당성이 있음이 분명하며 오히려 약간 더 좋은 해를 주고 있다고 볼 수 있다.

이번에는 paragon에서 CPU를 1개와 4개를 사용하여 계산하여 보았다. 그림 (c)와 (d)를 비교해보면 차이가 거의 없음을 알 수 있으므로 domain decomposition에 따른 오류는 없다고 볼 수 있다.

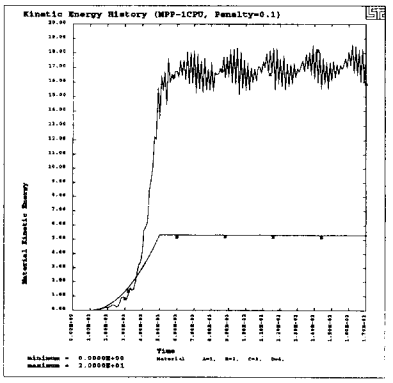
그림 3은 LS-DYNA3D 병렬 프로그램의 오류 중 하나였던 kinetic energy의 시간에 따른 결과이며 Cray에서의 결과와 대체로 일치하고 있다.



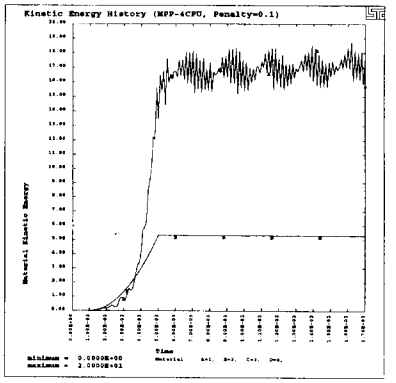
(a) Cray, Penalty=0.1



(b) Cray, Penalty=0.01



(c) MPP-1CPU, Penalty=0.1



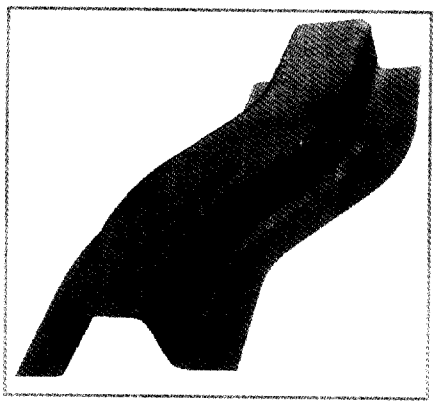
(d) MPP-4CPU, Penalty=0.1

그림 3 Kinetic energy history의 비교

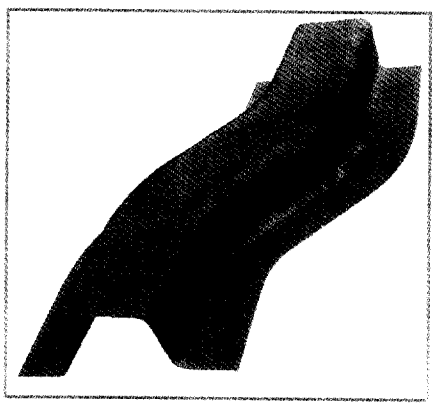
충격해석이나 충돌에서는 energy 관점에서의 결과가 상당히 중요하므로 반드시 고쳐져야 할 오류였으며 이제 완전히 해결되었다고 판단된다.

이상과 같이 stress와 energy의 비교로 판단할 때 paragon에서의 LS-DYNA3D 결과는 충분히 타당성이 있다고 볼 수 있다. 다만 같은 penalty parameter를 사용할 때 두 결과가 약간 틀린 이유는 접촉을 처리하는 알고리즘이 내부적으로 약간 다르기 때문이라고 생각되며 Cray에서 사용한 penalty parameter 값의 약 10배를 사용한다면 거의 일치하는 결과를 paragon에서 얻을 수 있을 것이다.

그림 4는 6000개의 shell 요소로 이루어진 박판성형 문제를 cray와 paragon에서 계산하여 성형 후의 두께분포를 보여주는 것으로서 32개의 CPU를 사용한 pagagon에서의 결과와 cray에서의 결과가 일치하고 있음을 보여준다.



(a) Cray-YMP



(b) Intel-PARAGON 32CPU

그림 4 박판성형 후의 두께 분포

2.2 병렬처리의 효율성

Paragon에서 얻어지는 결과가 타당성 있는 것이라 해도 계산시간을 고려할 때 Cray에 비하여 수행속도가 늦다면 사용자에게는 별로 유용한 프로그램이라고 할 수 없을 것이다. 그러므로 병렬처리 프로그램은, 다수의 CPU를 사용하여 계산을 수행할 때 효율적인 병렬처리가 가능하여 계산시간이 단축되어야 하며, 적절한 CPU 개수를 선정했을 때 Cray에서의 계산시간과 비교하여 비슷하거나 빨라야 한다.

다음의 두 표는 사용 요소의 개수가 다른 두 가지 문제를 paragon에서 CPU 개수를 증가시켜 가며 풀어낼 때 걸리는 시간을 비교한 것이며 괄호 안의 시간은 Cray에서의 CPU time을 나타낸다.

표 1 S-Rail 성형(그림 4 참조), 선형 shell 요소 6,000개

	4 CPU	8 CPU	16 CPU	32 CPU
CPU Time (13058 sec)	93202	56274	46071	34107
Cray대비 계산시간	7.14	4.31	3.53	2.61

표 2 Oil-Pan 성형, 선형 shell 요소 13,000개

	16 CPU	32 CPU	64 CPU	128 CPU
CPU Time (45130 sec)	88060	57280	37720	25830
Cray대비 계산시간	1.95	1.26	0.84	0.57

위의 두 결과에서, CPU 개수가 증가함에 따라 계산시간이 감소하는 것을 볼 수 있으며 특히 13,000개의 shell 요소를 사용하는 표 2의 경우 128개의 CPU를 사용하면 Cray 대비 거의 절반의 계산시간만으로도 결과를 얻을 수 있음을 알 수 있다. 더군다나 비교를 위하여 현재 사용하는 계산시간은 CPU time이므로, CPU를 공동으로 사용하여 여러 가지 작업을 동시에 수행하는 Cray의 경우, 실제로 user가 계산의 결과를 얻는 데 걸리는 시간은 CPU time의 수배 내지는 수십배가 될 수도 있기 때문에 paragon을 사용하면 훨씬 더 빠른 계산결과를 준다고 할 수 있을 것이다.

그러나 CPU 개수를 무조건 늘린다고 해서 계산시간이 그와 비례하여 계속 줄어드는 것은 아니다. 이것은 사용 CPU 개수가 늘어남에 따라 CPU 간의 상호 communication을 위한 message passing에 걸리는 시간도 같이 늘어나기 때문인데 이에 따라 사용자는 풀고자하는 문제의 크기에 따라 적절한 CPU 개수를 선정해야 할 것이다.

3. 맺음말

이상과 같이 paragon에서 DYNA3D 결과의 검증과 효율성을 점검하였는데, 적절한 CPU 개수를 선택하여 계산할 때는 Cray에 비하여 더욱 빠른 시간 내에 해석결과를 얻을 수 있을 뿐만 아니라 앞에서 언급되지는 않았지만 Cray에서는 풀 수 없는 아주 큰 memory를 요구하는 문제도 paragon의 DYNA3D를 이용하면 급속히 확산될 추세이며 이에 따라 기존의 응용 소프트웨어의 병렬화는 필연적인 결과라고 생각된다. 따라서 병렬컴퓨터 및 병렬 프로그래밍에 대한 이해를 통하여 미래의 시뮬레이션 분야에서 앞서갈 수 있는 능력을 갖추는 노력이 절실히 요구된다.

참 고 문 헌

1. Gallivan, K.A. et al., Parallel Algorithms for Matrix Computations, SIAM, Philadelphia, 1990.
2. Simon, H.D., "High performance computing in the U.S.-An analysis on the basis of the TOP 500 list."(unpublished).
3. 김진오, 김준태, "음향 수치해석의 초병렬 처리," 한국소음진동공학회지, 제6권 제4호, pp. 410-413, 1996
4. 2D Draw Bending, Proceedings of The 2nd International Conference NUMISHEET '93, 398-405, 1993
5. K. Mattiasson, A.Strange, P.Thilderkvist, A. Samuelsson, "Simulation of springback in sheet metal forming", NUMIFORM '95, USA, 115-124, 1995