

우리 生活을 바꿀 25가지 大發明 ①

# 새로운 世代의 컴퓨터

수잔 웨스트

미국과학진흥협회(AAAS)가 발행한 '사이언스 85'는 창간 6주년 특집으로 "다음 단계"(The Next Step)라는 제목으로 미래 우리의 생활을 바꿀 수 있는 25대 발견을 예고했다. '과학과 기술'지는 이 중에서 첨단기술분야부터 연재로 소개한다. <편집자>

종래의 컴퓨터는 한눈 팔지않고 혼자서 묵묵히 일하는 목수같은 것이었다. 못을 하나 하나 박고 나무 판자 한장한장 쌓아 올려가듯이 한번에 한단계씩 밖에는 문제를 처리할 수 없었다. 1940년대 이래 내내 이렇게 일해 왔던 것이다.

그러나 다시 태어나는 컴퓨터는 하나의 문제를 수백 또는 수천이라는 처리단위로 분할하여 이것을 동시에 짧은 시간내에 처리한 뒤 답을 토해 낼 것이다.

동시처리나 또는 병렬처리라고 불리는 이 방식은 아이디어로서는 별로 새로운 것은 못된다. 컴퓨터의 할아버지라고 할 수 있는 찰즈 바비지는 1840년대에 이미 이런 '분석기계'를 착상하고 있었다. 그의 '분석기계'를 사용하면 "동시에 여러개의 답을 낼 수 있게 조작할 수 있어 전체의 처리량을 크게 줄일 수 있을 것"이라는 생각이었다.

1947년 최초의 프로그램으로 움직이는 전자 계산기를 창조한 존 폰 노만도 병렬처리에 관심을 갖고 있었다. 그러나 이런 컴퓨터를 실현하려면 진공관, 배선을 포함한 전자부품의 비용은 엄청난 액수가 되기 때문에 이 아이디어를 포기해 버렸다.

대신 폰 노이만이 설계한 것은 3개의 독립된 부분으로 된 컴퓨터였다. 그것은 명령이나 데이터를 저장해 두는 기억장치(메모리), 이 명령에 따라 데이터를 처리하는 연산장치(프로세서) 그리고 이 2 장치의 교량역할을 하는 연결부(버스)이다. 이런 형의 컴퓨터에서는 계산처리를 할 때 마다 데이터나 명령을 메모리-버스-프로세서로 전달할 필요가 있다. 그리고 하나의 처리가 끝나면 이 결과는 프로세서로부터 메모리로 돌려주고 프로세서에는 또 다른 데이터 명령이 들어오게 된다. 이런 기제는 고속으로 만들 수 있으나 모든 데이터나 명령이 하나의 연결채널을 내왕하기 때문에 끝내는 망가지고 마는 것이다. 러시 아워의 고속도로와 같은 이런 상태는 폰 노이만의 '병목'으로 알려지게 되었다.

컴퓨터 과학자들은 이것을 해결하는 간단한 방법을 찾아냈다. 처음 트랜지스터가 등장하고 이어서 작은 실리콘 칩이 나와서 신호의 전달거리를 짧게 하고 전송시간을 절약했던 것이다. 칩의 크기가 작아지고 전송속도는 더 빨라지면서 고속 컴퓨터의 연산속도는 7년마다 한자리씩 올라갔다. 오늘날 슈퍼컴퓨터라고 불리는 이 컴퓨터는 매초 수백만회의 연산능력을 갖고 있

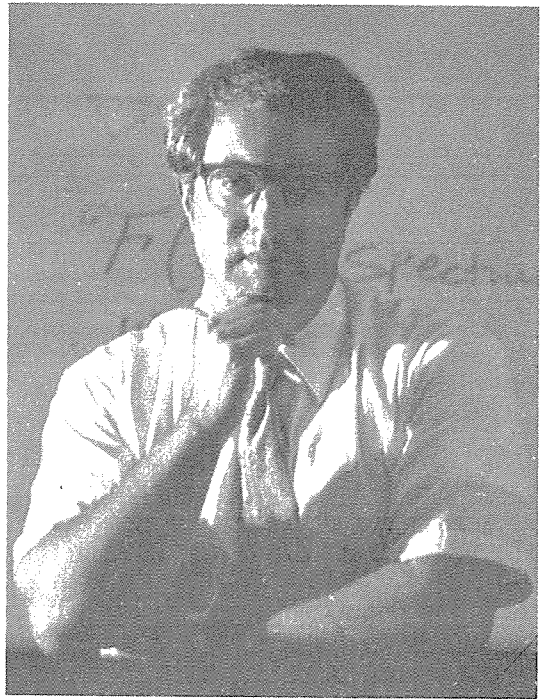
어 그 속도는 메가(1 백만) 플롭이라는 단위로 표시된다. 크레이 리서치사가 제작한 “크레이-2”의 연산속도는 1200메가플롭이다.

콘트롤 테이타사에서 나온 ETA시스템즈사는 1 만 메가플롭의 컴퓨터를 제작할 계획을 하고 있다. 그런데 퍼스널 컴퓨터를 플롭의 단위로 나타내면 가장 빠른 것도 0.05메가플롭에 지나지 않는다.

그러나 거의 모든 연구자들은 종래의 고속 컴퓨터 방식으로서의 대체로 이 정도가 속도의 한계라고 생각하고 있다. 물론 자잘한 것은 개량할 여지가 아직도 있다. 회로는 더욱 더 축소할 수 있을 것이며 이것을 냉각제속에 적셔주면 과열을 막을 수 있다. 실제로 크레이-2는 플루오리너트라고 하는 프레온과 많은 불화탄소로 냉각되고 있다. 칩은 보통의 실리콘이 아니고 칼륨 비소와 같은 반도체로 대체되어 전기신호의 전달시간을 단축할 수 있다. 그러나 이런 개량도 결국은 벽에 부딪치고 말 것이다. 전기신호를 빛보다 더 빨리 전달할 수는 없기 때문이다. 미국 슈퍼컴퓨팅연구센터의 시스템연구 부장이며 전 미국표준연구소의 컴퓨터 전문가였던 존 리가나티는 “우리는 물리학의 한계에 직면하고 있다”고 말하고 있다.

그래서 7년마다 10배 또는 그 이상의 페이스를 유지한다는 것은 처음부터 다시 시작해야 할 것 같다. 이제 전자부품은 값이 훨씬 싸졌기 때문에 컴퓨터 과학자들은 바비지와 폰 노이만이 할 수 없었던 것을 할 수 있게 되었다. 곧 많은 프로세서를 갖춘 기계를 제작하여 복잡한 일의 여러 부분을 동시에 처리함으로써 회답시간을 단축할 수 있게 된 것이다.

그런데 컴퓨터를 그렇게까지 고속화해야 할 이유는 어디에 있을까? 실상 컴퓨터는 오늘보다 더 빨라야 할까? 미국인들은 더 빠른 비행기를 좋아하는 것처럼 스피드에 대한 욕심이 많은데 바로 그런 때문일까? 그러나 컴퓨터의 능력을 올린다는 것은 실용적인 가치가 있으며 과학자의 연구에도 큰 관계가 있기 때문이다. 거의 모든 연구에서 과학자들은 우선 아이디어 또



인생에는 귀찮은 문제가 3개 있다. 죽음, 세금, 병렬처리컴퓨터이다. 우리는 이중 어느 하나도 피할 수 없다. (존 리가나티)

는 모델을 만든다. 소립자물리라면 쿼크, 해양학이라면 조류의 흐름, 경제학 이라면 GNP의 거동을 나타내는 모델이 될 것이다. 모델은 많은 수치나 방정식을 사용하여 기술한다.

연구자가 컴퓨터를 필요로 하는 것은 많은테이타를 처리하기 위한 것이지만 테이타가 많으면 많을수록 모델은 그만큼 충실하게 현실을 반영하게 된다. 그래서 모델에 따라서는 수천개의 방정식, 수천개의 변수를 갖는 것도 있다. 더우기 모든 방정식은 서로 일치해야 하기 때문에 몇번이라도 계산을 되풀이 해서 잘못을 최소한도로 줄일 필요가 있다. 테이타나 관련 물리법칙이 늘어나면 늘어날수록 컴퓨터의 작업량은 늘어난다. 이런 것은 모두 시간과 관련된다. 3차원의 해양모델을 예로 든다면 각 차원(세로, 가로, 깊이)의 테이타. 포인트를 2배로 늘이면 컴퓨터의 처리시간은 8배가 되어버린다.

또 테이타 포인트를 2배로 하면 날씨예보는 더욱 정확해 지겠지만 내일의 일기예보를

내는데 있어서 24시간이나 걸린다면 쓸모가 없는 예보가 되어 버린다. 미국 드라마일 섬의 원자력발전소 사고의 진행상황을 시뮬레이션하자면 오늘날의 슈퍼컴퓨터를 가지고도 5~10시간이 걸린다. 이렇게 되면 긴급시에는 쓸모 없는 것이 된다.

그런데 초보적인 병렬처리 라면 폰 노이만형 컴퓨터로서도 처리하고 있다. 비즈니스용의 미니컴퓨터는 많은 작업을 동시에 처리하고 있는 것처럼 보인다. 그러나 실제로는 작업과 작업사이를 신속하게 스위치를 점멸하고 있는데 지나지 않는다. (퍼스널 컴퓨터의 경우 이 능력은 멀티타스킹이라고 불린다) 1970년대에 크레이사와 콘트롤 데이터사가 벡터라고 불리는 병렬처리 기술을 채용했다.

컴퓨터는 서로 다른 많은 데이터에 대해 같은 처리를 여러번 되풀이 해야할 경우가 있다.

이 두 회사의 컴퓨터는 같은 일을 되풀이 하는 대신 벡터라는 일람표로 데이터를 종합하여 일괄 처리해 버린다.

이 두 회사는 최근 진정한 병렬처리 컴퓨터의 가능성을 모색하기 위해 2~4대의 프로세서를 연결했다. ETA의 기체는 8대의 프로세서를 연결했다.

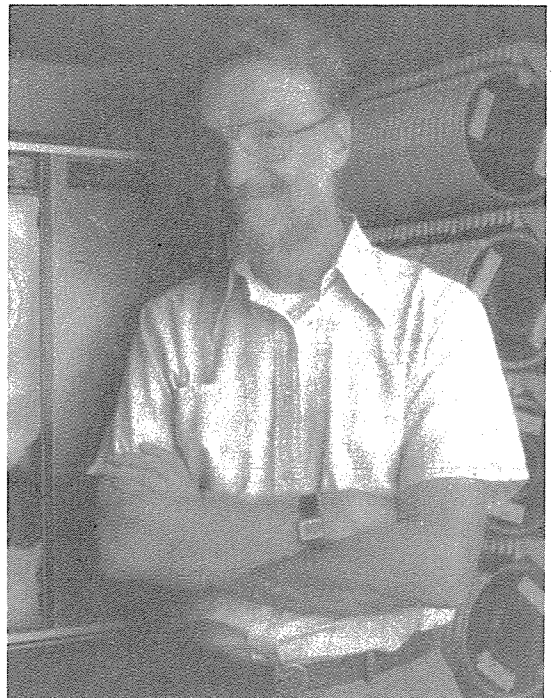
### ◇프로세서의 大集合

그러나 이런 기체는 컴퓨터의 재생을 겨냥하는 미국 일리노이대학의 데이비 쿡을 비롯한 많은 컴퓨터과학자들이 제작하기를 바라고 있는 멀티프로세서형은 아니다. 이들의 머리속에 있는 것은 수백대, 수천대의 프로세서의 집합이어서 4대나 또는 8대와의 자리수가 다른 것이다. 더우기 한대에 수백만달러나 하는 크레이나 콘트롤 데이터사의 초고속컴퓨터를 몇대만 연결한다고 해도 막대한 비용이 들고 작업을 불과 몇개 분할하는데 그친다면 한계가 드러나 보인다. 한대 한대의 능력은 떨어지지만 보다 짝 프로세서를 여러개 연결하여 되도록 잘게 분할한 문제를 처리시키는 편이 훨씬 현명하다고 새

로운 컴퓨터개발자들은 말하고 있다.

그러나 이런 기대를 실현하려면 컴퓨터 제작자들이 일찍이 직면한 일이 없었던 많은 어려운 문제를 해결해야 한다. 한마디로 프로세서의 문제라고 할 수 있다. 이렇게 많은 프로세서간에 교환되는 명령이나 데이터의 전송으로 시간을 낭비하지 않게 하기 위해서는 시스템을 어떻게 관리할 것인가? 모든 프로세서를 하나의 집약된 거대 메모리에 액세스시켜야 할까? 또는 프로세서에 독자적인 소형 메모리를 갖게 하는 것이 좋을까? 하나하나의 프로세서가 모두 서로 '대화'를 하는 것과 바로 옆의 프로세서끼리만 '대화'를 하는 것은 어느쪽이 효율적일까? 또 모든 프로세서를 효율적으로 작동할 수 있게 작업을 세분할 수 있을까?

이런 일련의 의문에 답하는 것은 루빅큐브를 해결하는 것과 다르다. 큐브는 비틀때 마다 답은 더 멀어지는 것처럼 보인다. 그런데 각 프로세서에 전용 메모리를 준다고 하자. 이런 경



〈데이빗 쿡〉

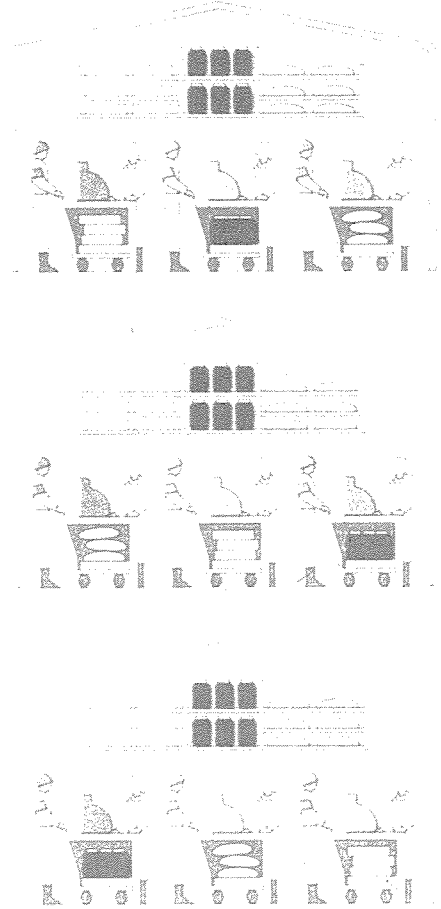
지금보다 몇백만배나 빠른 컴퓨터가 있으면 과학자나 기술자들은 하고 싶은 연구는 무엇이든지 할 수 있다.

## 군대의 쇼핑을 예로 든 컴퓨터의 기능차이

축차처리형



기본병렬형

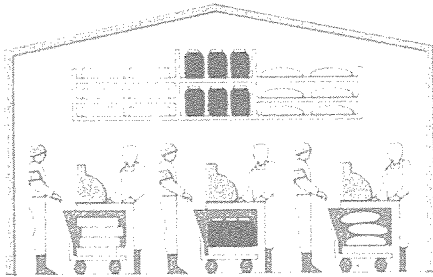
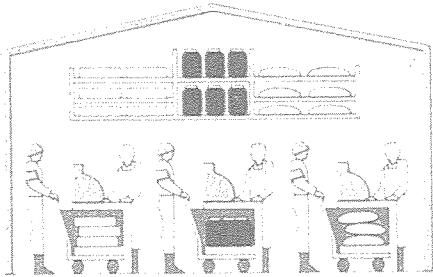
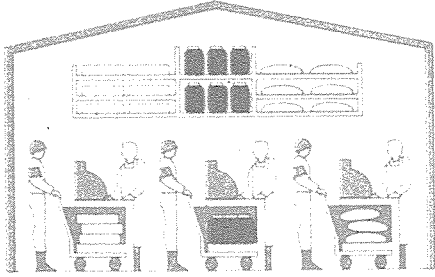


컴퓨터의 축차처리, 병렬처리는 쇼핑에 비유할 수 있다. 군인 60인분의 저녁식사의 재료를 사는 경우를 생각해 본다. 1인당 스파게티 1상자, 스파게티 소스 1깡통, 빵 한덩어리를 먹는다고 한다. 스파게티, 스파게티 소스, 빵은 컴퓨터에 입력하는 데이터에 해당된다. 군인은 각자가 쇼핑하도록 명령을 받는다. 프로그램을

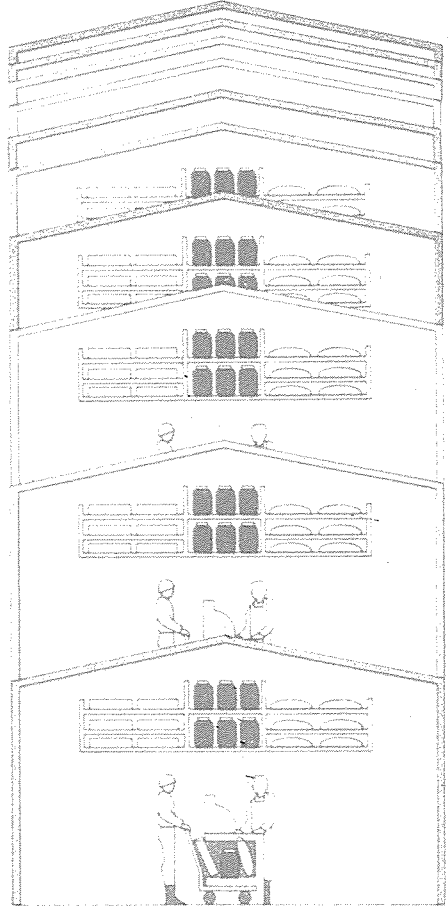
실행하는데 메모리에 명령을 보내거나 메모리에서 데이터를 끄집어 내는 것이나 같은 일이다. 단순한 축차형 컴퓨터에서는 프로세서는 데이터를 하나씩 메모리에서 끄집어 내어 이것을 처리하여 답을 내는 작업을 되풀이 한다. 축차처리형은 60명의 군인이 한사람씩 상점으로 들어간다. 스파게티, 소스, 빵을 집을 때마다 돈을 지불한다. 금전지불소에서 계산을 아무리 빨리 한다고 해도 군인 한사람씩 식품을 선반에서 카운터까지 들고 오는데는 시간이 걸린다. 축차형 컴퓨터의 속도도 프로세서가 데이터를 기다리는 시간으로 제약을 받는다.

병렬형 컴퓨터방식으로 쇼핑하면 훨씬 빨라진다. '기본병렬형'의 조보적인 병렬형 컴퓨터에서도 일을 분할하여 1대 이상의 프로세서에 배

고급병렬형



궁극병렬형



분한다. 여기서는 60인의 군인이 20명씩 3개의 그룹(군인의 완장색으로 분별)으로 분할하여 3개의 카운터를 갖춘 상점에 나가 그룹마다 카운터 하나를 전용으로 사용한다. 1개의 그룹에서 선발된 한사람의 군인은 우선 스파게티 20상자를 선반으로부터 집어내어 계산소에서 돈을 지불한다. 각 계산소는 20인분의 계산을 3회씩 되풀이한다. 컴퓨터에서는 백터라고 불리는 처리방법이지만 1인분씩 계산하는 것보다는 분명히 빠르다.

그러나 3개의 그룹의 군인이 동시에 스파게티를 끄집어 내는 경우가 어떻게 될까? (여러대의 프로세서가 같은 데이터를 동시에 요구하는 경우). 교통정리가 끝날때까지 쇼핑은 일사 중단될 수 밖에 없다. 그래서 고급병렬형의 보

다 앞선 컴퓨터에서는 프로세서끼리 박치기를 하지 않게 메모리를 갖고 있는 것이다. 군인의 그룹마다 전속의 상점을 갖고 있는 것과 같다. 그룹 대표 3인이 전용점에 가서 한사람은 스파게티, 한사람은 소스, 한사람은 빵을 각 20인분씩 왜건에 담아 각각 전용의 카운터에서 돈을 지불한다.

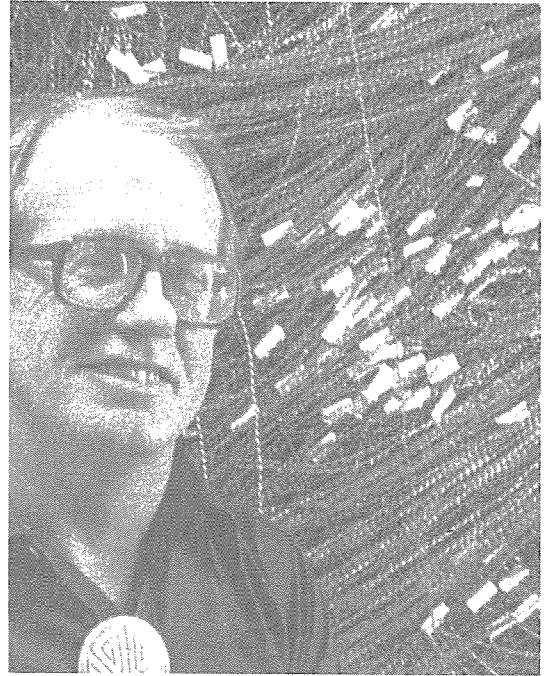
그러나 병렬처리중에서 가장 빠른 것은 하나의 작업을 1대의 프로세서로 처리할 수 있게 조정된 뒤에 일제히 처리하는 방법일 것이다. 이 '궁극병렬형'의 쇼핑에서는 군인 각자가 전속의 상점을 갖고 그곳에서 모두 동시에 산다. 이런 완전한 병렬처리 컴퓨터는 제작비가 엄청나게 먹혀 당분간은 컴퓨터과학자들의 꿈속에서 맴돌고 있다.

우에는 어떤 프로세서가 다른 프로세서의 메모리에 들어있는 정보를 필요로 하는 경우 전송에 시간이 걸린다. 많은 프로세서를 도입하여 얻은 시간은 결국 모두 잡아먹게 되는 것이다. 그렇다면 거대 메모리 하나를 모든 프로세서가 공유하게 하면 어떻게 될까? 이렇게 되면 1대 이상의 프로세서가 같은 데이터를 필요로 하게되는 경우의 해결책을 마련해야 한다.

프로세서의 작업을 고루 돌아가게 쪼개는 일도 쉽지 않으며 프로세서간의 명령을 전송하는 방식의 문제도 있다. 예컨대 프로세서를 크리스머스 큐리와 같이 배치하고 각 프로세서의 정보전송을 상위 프로세서 1대, 하위 프로세서 2대간으로 제한한다고 하자. 이때 큐리 반대편이 프로세서에 무엇을 보내야 할 경우에는 반드시 큐리의 정상을 경유해야 하며 상당한 시간이 낭비된다. 이렇게 프로그램에 의한 직업분할과의 균형문제도 있고 해서 프로세서가 많을수록 좋은 것도 아니다.

미국 컬럼비아대학의 데이빗쇼와 그의 동료 과학자들은 나무모양으로 배치하는 방법을 연구하고 있다. 이들이 구상하고 있는 원형의 '비(非) 폰 노이만' 식 기계는 결국 1백만대의 프로세서를 가질 예정이지만 지금까지 연결한 것은 63대에 지나지 않는다.

뉴욕대학의 극초컴퓨터 프로젝트의 책임자인 앨런 고트리브는 4096개 프로세서의 메모리 공유형기계 설계를 지휘하고 있다. 오스틴의 텍사스대학에서 개발하고 있는 "텍사스 반복 컴퓨터배열"은 카멜레온형이라고 불린다. 어떤 프로세서와 어떤 메모리가 교신할 것인가는 프로그램에 달려있을 뿐 아니라 폰 노이만형은 물론 병렬 기계로서도 사용할 수 있다. 캘리포니아공대의 '코스믹 큐브'는 2년전부터 작동하고 있으나 64대의 프로세서를 다차원의 큐브가이 배열한 기계이다. 일본에서는 정부자금에 의한 프로젝트가 개수는 적으나 초고속의 프로세서를 조합한 고속 범용컴퓨터를 개발하고 있다. 어떤 그룹이 먼저 승리를 거둘 것인지 아직 아무도 모른다.



컴퓨터의 고속화로 복잡하고 정확한 모델을 만들 수 있다. (빌 버즈비)

### ◇선보일 '시더 컴퓨터'

데이빗 쿡은 이미 15년전에 '시더'라고 불리는 컴퓨터를 머리속에 그리고 있었으나 지금까지 제작을 미뤘다. 이제 시기가 무르익었다. 쿡의 팀은 프로그램은 프로세서간의 조정을 좌우하는 열쇠라고 믿고 있으나 이런 견해는 컴퓨터계에서는 조금 이단적인 생각이다. 그러나 이들은 종래의 컴퓨터개발 방법에 대해 가차없는 비판을 하고 있다.

"컴퓨터 메이커들은 기계를 만든 뒤 소프트웨어를 생각하거나 또는 전혀 생각도 하지 않아 우리들을 당혹하게 만든다. 누구나 컴퓨터의 능력을 모두 이용할 수 없는 것은 바로 이런 이유 때문이다. 우선 컴퓨터의 회로와 하드웨어를 고정해 버린다면 정해진 일밖에 할 수 없는 것은 뻔한 일이다"고 그는 말하고 있다.

일리노이대학의 그룹은 다른 컴퓨터 과학자들과는 달리 동시처리를 위한 전용 컴퓨터 언어가 필요하다고 생각하지 않는다는 점에서 특

이다. 과학기술용의 표준컴퓨터 언어인 포트란으로 쓰여진 종래의 프로그램을 버릴 필요가 없다는 입장이다. '파라프레이즈'라고 불리는 이들의 소프트웨어에서는 전통적인 포트란의 프로그램을 세분화 하여 되도록 병렬처리에 접근시킨다. 종래의 프로그램의 이곳저곳을 서로 관련시킨 것이어서 예컨대 어떤 명령을 수행하는데 프로그램의 다른 부분에서 발생하는 데이터가 필요한 경우가 있다. 프로그램의 명령을 하나하나 처리해 나가는 재래식 기계라면 이런 것도 상관이 없다. 그리고 프로그램 전체를 한번에 처리하게 설계된 컴퓨터의 경우는 곤란하다. 파라프레이즈에서는 우선 프로그램의 독립된 부분을 발견하여 각각 독립된 작업으로서, 동시처리 프로세서에 각각 입력시킨다. 파라프레이즈가 분할할 수 없는 프로그램과 만나면 이 프로그램에 대해서도 더욱 상세한 정보를 요구하고 다시 시도해 본다. 한마디로 프로그램의 분할이라고 하지만 '알고리즘'(데이터로 부터 필요한 답을 얻기 위한 순서)을 고쳐 써야 하는 일이 포함될 수도 있고 데이터를 백타로서 고쳐야 할 경우도 있다. 처리순서를 모두 바꿔야 할 필요가 있는 경우도 있다. 마지막으로 시더는 8대의 프로세서로 된 클러스터(뭉치)가 많이 모여 구성된다. 프로세서마다 전용메모리를 갖게 되나 동시에 덩어리 수준에도 메모리를 장비한다. 프로세서는 하나하나의 프로세서간을 교신할 수 있을 뿐 아니라 상위의 거대 메모리로 부터도 자유롭게 정보를 끄집어 낼 수 있다. 이 고속 스위치는 쿡의 공동연구자인 던칸 로리가 설계했다.

시더 성능의 열쇠가 되는 이 스위치는 수파게티처럼 서로 얽힌 회선을 헤치면서 전화를 연결하는 크로스 바 스위치와 닮았다. 하나 하나의 클러스터는 쿡과 밀접한 협력을 하고 있는 보스턴의 얼라이언트 컴퓨터 시스템사가 납품한 것이며, 실제로는 프로세서 8대를 갖춘 소형 컴퓨터라고 할 수 있다. 클러스터의 연산속도는 95메가플롭이다.

하나의 클러스터에 고르게 일을 분배하는 것

은 소프트웨어의 역할이다. 클러스터 내부에서는 배당된 일을 가장 먼저 끝낸 프로세서로 부터 순서적으로 다음 일을 할당한다. 또 이 소프트웨어에서는 데이터나 명령을 분할하여 3단계로 제충화된 각 메모리에 기억 시키고 있다. 교신에 의한 시간의 낭비를 최소한으로 줄이기 위한 것이다. 또 2대의 프로세서가 동시에 같은 데이터를 처리 하는 것을 막기 위한 방법도 고려하고 있으며 아직도 해야 할 일은 산더미처럼 많다. 2개의 클러스터부터 시작하여 1989년에는 16클러스터(128대의 프로세서)로 늘릴 계획이다. 쿡의 공동연구자인 사메는 그동안 늘어나는 프로세서에 일을 효율적으로 배분하게 알고리즘을 고쳐써야 한다. 10년 뒤 연산 속도 1백만 메가플롭의 시더가 실현되기를 쿡은 기대하고 있다.

그러나 병렬처리형 컴퓨터가 스피드에 대한 소원을 만족시켜 주리라는 보증은 없다. 여러 아이디어를 검증해 보려고 해도 현실적으로 움직이고 있는 이런 형의 컴퓨터는 아직 1대도 없다. 시더와 같은 컴퓨터가 종래의 것보다 어느 정도 빠르고 주어진 문제를 처리할 수 있을까, 사실은 아무도 알지 못하는 것이다. 또 어떤 수학적인 과제도 하나 하나의 프로세서에 효율적으로 분배할 수 있는 분할이 가능한가는 것도 알 수 없다. 시스템을 조정하기 위한 추가 프로그램 때문에 많은 프로세서의 도입으로 절약된 시간을 소비해 버리는 수도 있을 것이다.

염려되고 있는 것은 이런 문제뿐 아니다. 로스알라모스 국립연구소의 빌 버즈비는 일 전체를 병렬처리하지 않으면 동시처리 시스템의 속도가 대단한 것은 아니라는 사실을 입증하였다. 예컨대 일의 반만을 동시처리한다면 16대의 프로세서로 구성된 시스템의 스피드는 프로세서 1대의 경우보다 겨우 2배 빠른데 지나지 않는다. 동시처리의 비율을 95%로 올린다고 해도 9배 정도 빠르다는 것이다. 컴퓨터 과학자로서는 이것은 중대한 도전이다. 그러나 병렬처리가 유일한 길이라는데 이론을 제기하는 연구자는 없다. (玄 源 福 譯)