13장 멀티프로세서

M. Mano



Demand for Computational Speed

Grand Challenge Problems

- Areas requiring great computational speed include numerical modeling and simulation of scientific and engineering problems
- Modeling large DNA structures,
- Global weather forecasting,
- Modeling motion of astronomical bodies.

Weather Forecasting

- Whole global atmosphere divided into cells of size 1 mile x 1 mile x 1 mile x 1 mile to a height of 10 miles (10 cells high) about 5 x 10⁸ cells.
- Suppose each calculation requires 200 floating point operations. In one time step, 10¹¹floating point operations necessary
- To forecast the weather over 10 days using 10-minute intervals, a computer operating at 100 Mflops (10⁸ floating point operations/s) would take 10⁷ seconds or over 100 days.
- To perform the calculation in 10 minutes would require a computer operating at 1.7 Tflops (1.7 x 10¹² floating point operations/sec).

13.1 멀티프로세서의 특징

- 멀티프로세서 시스템
 - 메모리, I/O 장치 등을 공유하는 2개 이상의 CPU를 가진 시스템
 - 하나의 운영체제에 의해 프로세서와 시스템의 다른 요소들 사이의 상호 동작이 제어되며, 동일 문제 해결을 위해 모든 요소들이 협력
 - 멀티컴퓨터
 - 여러 개의 컴퓨터가 통신라인을 통해 연결되어 컴퓨터 망을 형성, 각 컴퓨터들은 독자적으로 동작하고 통신함 (Cluster Computing)
- 병렬 처리로 인한 성능 향상
 - 1) 다수의 독립된 작업들이 병렬적으로 수행
 - 2) 하나의 작업이 여러 부분으로 나누어져 각각이 병렬적으로 수행
- 시스템의 신뢰성(reliability) 향상
 - 한 부분의 고장, 에러가 다른 부분에 미치는 영향이 제한적

멀티프로세서의 분류

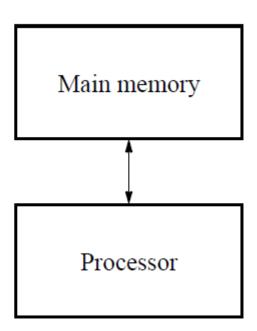
- 공유 메모리 멀티프로세서 (shared memory or tightly coupled multiprocessor)
 - 공통의 공유 메모리를 가짐, 모든 CPU가 메모리주소 공유
 - 각 프로세서는 로컬 캐시(local cache) 보유 가능
 - 장점: 프로그램 용이, 별도의 통신 메커니즘 불필요
 - 단점: 통신 지연 (memory ~ Processor)
- 분산 메모리 시스템 (distributed memory or loosely coupled system)
 - 각 프로세서들은 각각 **개별적인 로컬 메모리**를 가짐
 - 프로세서들은 교환(switching)구조에 의해 연결
 - 메세지 전달(message-passing)방식의 정보 이동
 - 장점: 빠른 local processing
 - 단점: 프로그래밍이 어렵다
 - Cf) 클러스터(Cluster) 컴퓨팅, 그리드(GRID) 컴퓨팅

Conventional Computer

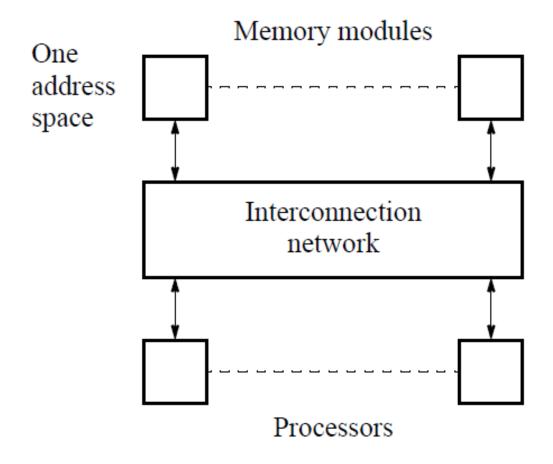
Single processor and memory

Instructions (to processor)

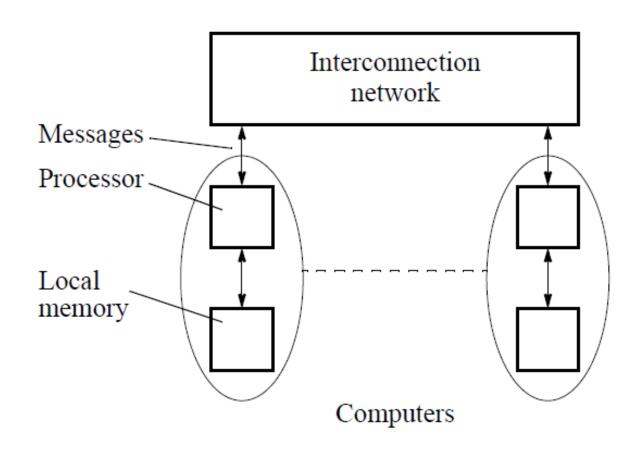
Data (to or from processor)



Shared Memory Multiprocessor







클러스터 컴퓨팅

■ 컴퓨터 클러스터(computer cluster)는 여러 대의 <u>컴퓨터</u>들이 연결되어 하나의 시스템처럼 동작하는 컴퓨터들의 집합을 말한다. 클러스터의 구성 요소들은 일반적으로 고속의 <u>근거리 통신망으로 연결된다. 서버로 사용되는 노드에는 각각의 운영 체제</u>가 실행된다. 컴퓨터 클러스터는 저렴한 <u>마이크로프로세서</u>와 고속의 <u>네트워크</u>, 그리고 고성능 <u>분산 컴퓨팅</u>용 소프트웨어들의 조합결과로 태어났다. 클러스터는 일반적으로 단일 컴퓨터보다 더 뛰어난 성능과 안정성을 제공하며, 비슷한 성능과 안정성을 제공하는 단일 컴퓨터보다 비용 면에서 훨씬 더 효율적이다. 따라서 열개 안팎의 중소 규모의 클러스터부터 수천 개로 이루어진 대형 <u>슈퍼컴퓨터</u>에 이르기까지 널리 사용되고 있다.

(Wikipedia)

클러스터 컴퓨팅



독일 <u>켐니츠 기술 대학교</u>, 대규모 <u>리눅스</u> 클러스터

Grid Computing

■ 그리드 컴퓨팅(grid computing)은 최근 활발히 연구가 진행되고 있는 분산 <u>병렬 컴퓨팅</u>의 한 분야로서, <u>원거리 통신망</u>(WAN, Wide Area Network)으로 연결된 서로 다른 기종의 컴퓨터들을 하나로 묶어 가상의 대용량 고성능 컴퓨터(super virtual computer)를 구성하여 고도의 연산 작업(computation intensive jobs) 혹은 대용량 처리(data intensive jobs)를 수행하는 것을 일컫는다. 모든 컴퓨터를 하나의 초고속 네트워크로 연결하여 계산능력을 극대화시키는 차세대 <u>디지털 신경망 서비스</u>를 말한다.

(Wikipedia)

SETI@Home





Data info

Name: WarLokK Team: L'Alliance Francophone Total arealt: 0,00

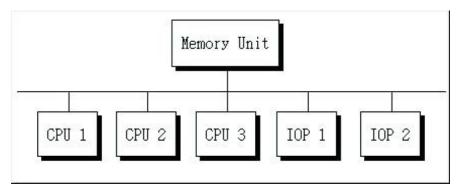
SETI@Home Enhanced

13.2 상호연결망 구조

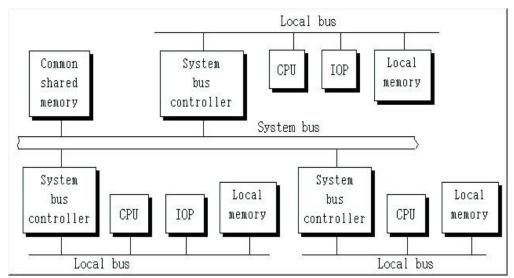
- Interconnection networks
- 상호연결망의 형태 (Topology)
 - 시분할 공통 버스
 - 링
 - 트리
 - 다중포트 메모리
 - 크로스바 스윗치
 - 다단 교환망
 - 하이퍼큐브 시스템
 - 토러스 (Torus)

시분할 공통 버스

- 공통 버스:
 - 통신 효율성 감소

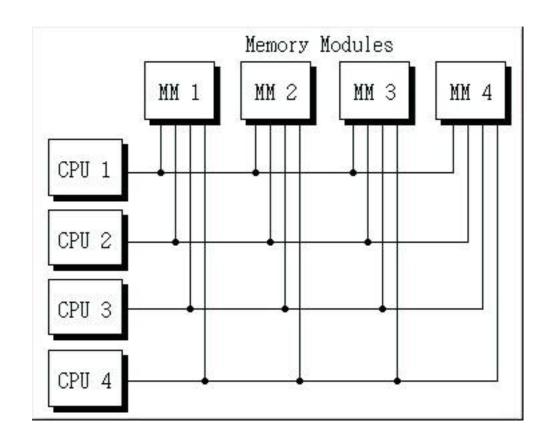


- 이중 버스 구조:
 - 계층적 구성

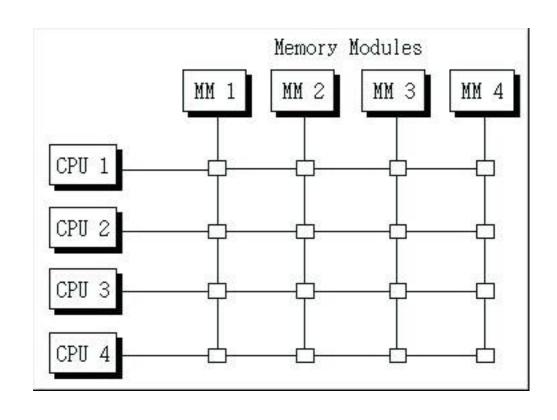


다중 포트 메모리

- 다중 통로에 의한 높은 전송률
- 비용 증가



크로스바 스위치



다단 교환망

- 기본 요소:
- 2입력, 2출력 스위치

A connected to 0

A connected to 1

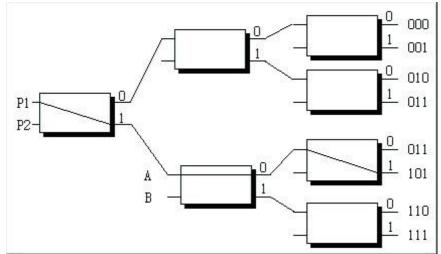
A connected to 1

A connected to 1

B connected to 0

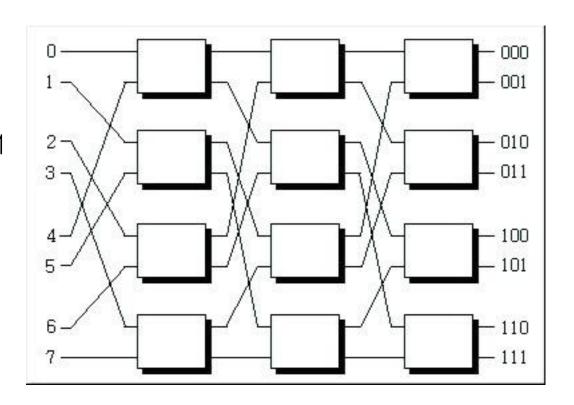
B connected to 1

• 2x2 스위치로 구성된 이진 트리 형태의 망



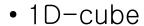
다단 교환망

- 8x8 오메가 교환망
- source routing ex) destination: 011



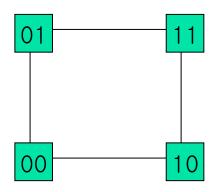


하이퍼 큐브

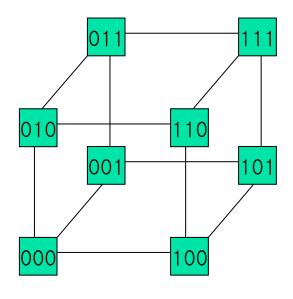




• 2D-cube

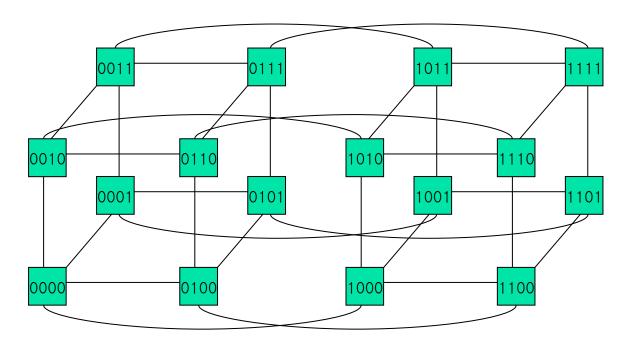


• 3D-cube



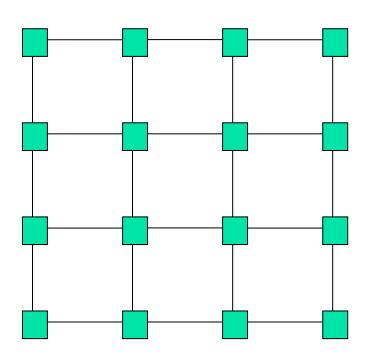


4D-cube

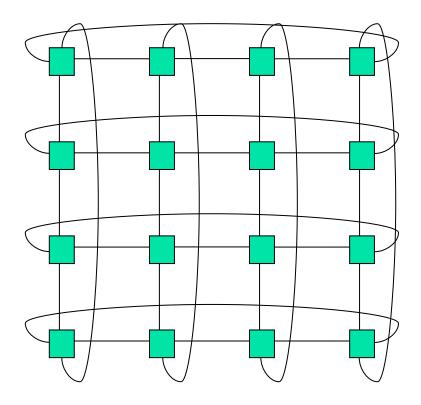




• 4x4 Mesh

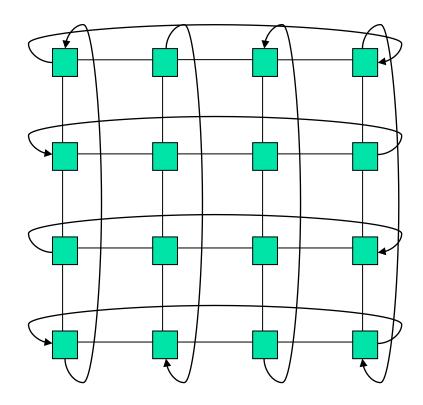


• 4x4 Torus



MSN

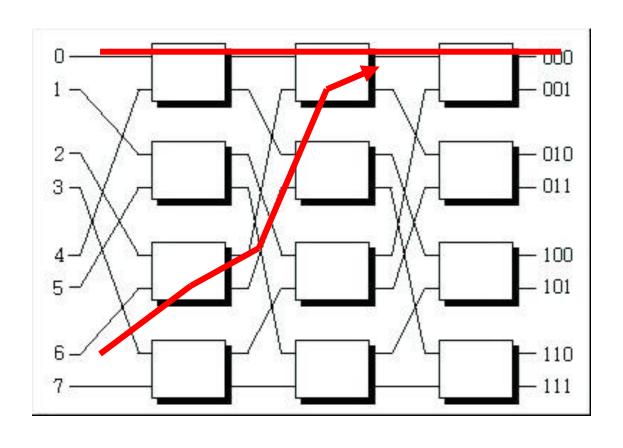
Manhattan Street Network



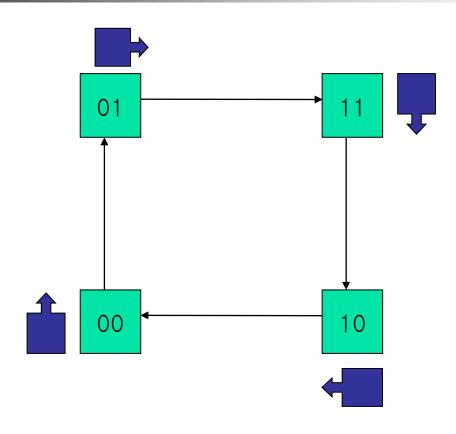
상호연결망 비교

Network	node degree	diameter	bisection width
Linear array	2	N-1	1
Ring	2	N/2	2
Binary Tree	3	2(h-1)	1
Star	N-1	2	N/2
2D Mesh (rxr)	4	2(r-1)	r
2D Torus (rxr)	4	2(r/2)	2r
Hyper cube	n	n	N/2
Complete grap	oh N−1	1	$(N/2)^2$

Blocking







13.3 프로세서간 중재

- 동적 중재 알고리즘
 - 시스템이 동작하고 있는 동안 장치들의 우선순위를 변경시킬 수 있는 기능 제공
- 시간 분할(time slice) 알고리즘
 - Time Slot을 라운드-로빈(round-robin)방식으로 각 프로세서에 할 당
- **폴링**(polling)을 이용한 버스 시스템
 - 폴(pole) 라인으로 버스 승인(grant) 신호를 대체하여 모든 장치에 연결, 이 라인을 이용하여 버스 제어기가 각 장치의 주소를 결정하 게 한다
- LRU(least recently used) 알고리즘
 - 가장 오래동안 버스를 사용하지 않은 장치에게 가정 높은 우선순위를 부여
- first-come first-serve 알고리즘
 - 요구가 도착한 순서대로 서비스 제공
- 회전 데이지 체인(rotating daisy chain) 알고리즘

13.4 프로세서간 통신과 동기화

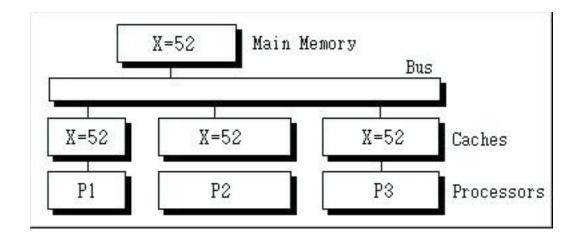
- 공유메모리 시스템
 - 송신 프로세서가 공유메모리에 데이터를 저장
 - 수신 프로세서는 폴링 또는 인터럽트 신호를 받고 데이터를 읽어서 처리
- 분산 메모리 시스템
 - 메시지 패싱 방식으로 프로세서간 통신
- 멀티프로세서에서의 운영체제 방식
 - Master-Slave: 주 프로세서에서만 운영체제 동작
 - Separate: 각 프로세서가 독립적인 운영체제
 - Distributed: 하나의 운영체제가 분산되어 있음

상호배제 (mutual exclusion)

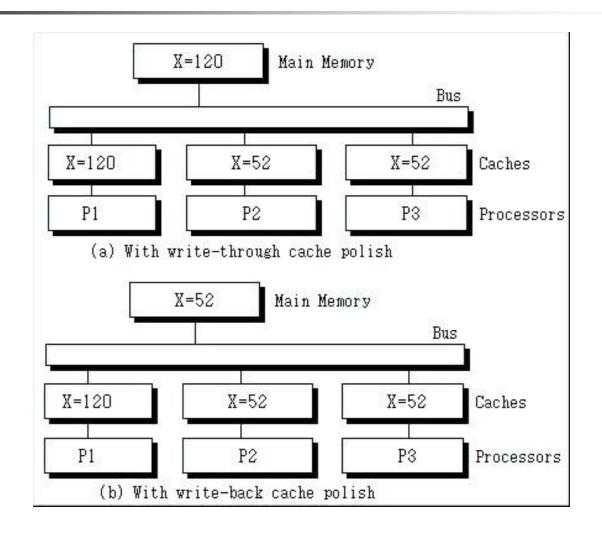
- 상호배제
 - 두개 이상의 프로세서에 의하여 데이타가 동시에 변화되는 것을 방지하기 위해 공용의 메모리나 다른 자원에 차례대로 접근할 수 있도록 하는 기법
 - 한 프로세서가 임계 프로그램 영역(critical program section)에 있을 때 다른 프로세서가 접속하지 못함
 - 임계 프로그램 영역: 한번 시작하면 다른 프로세서가 같은 공용 자원에 접근하기 전에 실행을 완료해야 하는 프로그램 시퀀스
- 세마포(semaphore)
 - 프로세서가 임계 프로그램 영역을 수행하고 있는 지의 여부를 표 시하는 2진 변수
 - 세마포 = 1: 한 프로그램이 임계영역을 수행하고 있음
 - 세마포 = 0: 다른 프로세서에게 공용 자원의 할당이 가능함
 - Test and Set: R ← M[SEM] test, M[SEM] ← 1 set

13.5 캐쉬 일관성

- 멀티프로세서에서 각각의 캐쉬 데이터는 일관성 유지가 필요함
- 읽기 전용 메모리는 일관성이 유지됨
- 쓰기 동작에서 일관성의 상실 발생



캐쉬 일관성



캐쉬 일관성 문제 해결

- 공통의 캐쉬 사용
 - 평균 접근시간 증가
- 캐쉬 제한
 - 공유가능 데이터는 캐쉬에 저장하지 않음
- Cache Coherence Protocol
 - 디렉토리 방법(directory-based)
 - 중앙집중식 표 이용, 데이터의 분산 상황과 Update 상황 표시
 - 메모리 블럭의 상태 저장, 각각의 블럭에 읽기전용(read only, RO)
 과 읽기 및 쓰기(read and write, RW) 표시
 - 스누핑(snooping)
 - 버스의 쓰기요구를 감시, local cache의 valid bit를 제어
 - write-through 정책 사용
 - 버스 하드웨어장치에 의한 제어