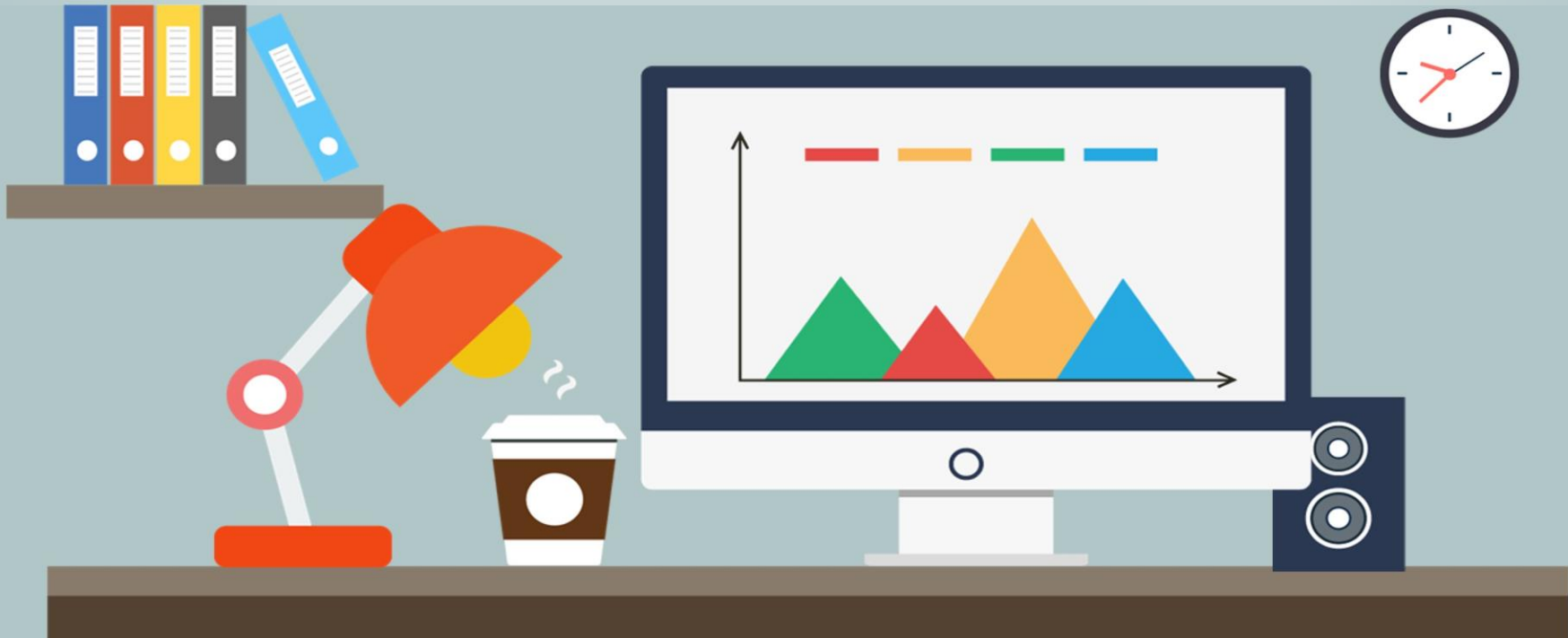


14주 3강

다중 프로세서 시스템





- 다중 프로세서 시스템을 구성하는 프로세서들이 동일한 성능으로 수행하고 동일한 기능을 포함하는 것
- 버스 또는 다른 상호 연결 망에 의해 프로세서들이 서로 연결된다.
- 공유된 기억장치와 채널공유
 - 각 프로세서가 액세스하는 데 걸리는 시간이 거의 동일하며, 모든 프로세서는 입출력장치의 액세스를 위한 채널을 공유한다.
- 하나의 통합 운영체제가 모든 프로세서와 입출력장치, 기억장치를 관리
 - 프로세서들이 하나의 운영 체계를 사용
 - 운영체제는 프로세서들 간의 상호작용(interaction)을 지원
 - 작업(job), 태스크(task), 파일(file) 및 데이터 요소 수준에서 프로그램들 간의 상호작용을 지원할 수 있다.

대칭형 다중프로세서의 장점

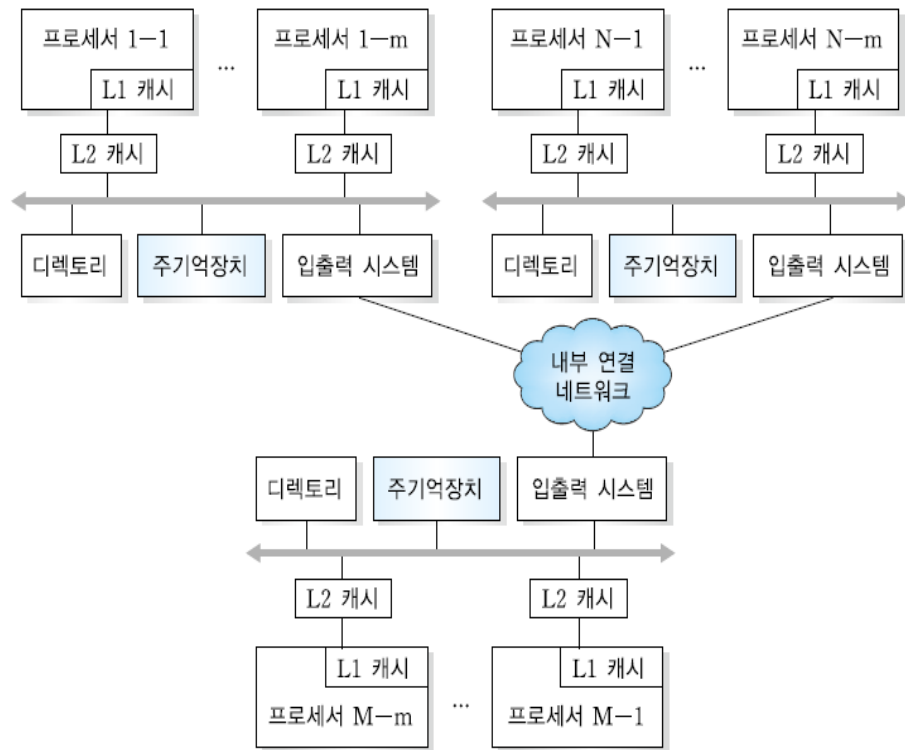


- 대칭 특성으로 작업 부하를 효과적으로 분산시킬 수 있으며 신뢰성도 높다.
 - 한 프로세서가 고장일 경우 운영 체제는 그 프로세서를 제거한 뒤 성능이 조금 떨어진 상태에서 계속 작동할 수 있다.
 - 한 프로세서에서 실행하던 작업을 다른 프로세서로 옮겨 실행할 수 있다.
 - 어떤 작업은 처리기가 협력하여 실행할 수 있다.
- 장점
 - 각 프로세서의 작업들이 병렬로 분산 처리되기 때문에 성능이 더 높아진다.
 - 어느 한 프로세서가 고장 나도 나머지 다른 프로세서들이 대신해서 작업을 수행할 수 있으므로 시스템의 중단 없이 계속 동작할 수 있다.
 - 프로세서들의 추가를 통해서 성능을 높일 수 있다. 그래서 점진적인 확장이 가능(Incremental growth)하다.
 - 제조 회사들은 프로세서 수에 따라 가격과 성능이 다양한 제품들을 제공할 수



- 불균일 기억장치 액세스(NUMA, Non-Uniform Memory Access) 시스템
 - 시스템 내의 모든 프로세서가 동일한 기억 장치를 공유하고 있지만 기억 장치를 접속하는 시간이 기억 장치의 위치에 따라 다르다.
 - 대칭형 다중 프로세서가 버스 병목 현상으로 프로세서의 수가 제한되는 것을 극복하기 위해서 등장
- 특징
 - 기억장치의 투명성을 제공하고 다수의 프로세서 노드 사용이 가능하다.
 - 각 노드는 버스를 통해서 지역 기억장치뿐만 아니라 고속의 상호 연결 버스를 통하여 원격 기억장치에 접속할 수 있다.
 - 모든 프로세서는 일정한 전역 기억장치를 공유하며, 접속하는 데 소요되는 시간은 기억장치의 물리적 위치에 따라 달라진다.
 - 인접한 곳에 위치한 기억장치에는 빠른 접속이 가능해서 우수한 시스템 성능을 발휘할 수 있다.
- 캐시 일관 NUMA(Cache-Coherent Non-uniform Memory Access)시스템
 - 여러 프로세서의 캐시들 사이에 캐시 일관성이 유지되는 NUMA 시스템이다.

CC-NUMA 조직의 일반적인 구성도



CC-NUMA 조직의 일반적인 구성



- 상호연결 망을 중심으로 일반 다중 프로세서 조직들이 각 입출력 시스템을 통해서 연결
 - 다중 프로세서 조직 내에 주기억장치를 가지고 있고, 각 프로세서는 2계층의 캐시를 가지고 있다.
- 해당 다중 프로세서 조직 내에서, 프로세서들이 공유된 해당 주기억장치에 액세스하는 속도는 거의 동일하다.
- 다른 다중 프로세서 조직에 있는 주기억장치에 액세스하려면 상호연결 망을 통해서 액세스
 - 다중 프로세서 조직내의 액세스 속도보다는 느리다.
 - 결과적으로 어느 영역을 액세스하느냐에 따라서 액세스 속도 또는 시간이 다르다.
- 각각의 프로세서에 연결되어 있는 캐시들은 각각의 주기억장치와 일관성을 유지한다.



- 소프트웨어를 거의 변경하지 않아도 대칭형 다중 프로세서(SMP)보다 더 높은 수준의 병렬성을 이용하여 효과적인 성능을 제공할 수 있다.
- 원격 노드에 대한 기억장치 액세스로 인한 성능저하를 피할 수 있다.
 - 캐시의 사용은 원격 액세스를 포함하여 기억장치 액세스를 최소화할 수 있다.
- 운영체제와 응용을 대칭형 다중 프로세서(SMP)에서 CC-MUMA로 이동시키기 위해서는 소프트웨어를 변경해야 한다.
 - 시스템을 구성방법에 따라 가용성 문제가 발생할 수 있다.

클러스터(Clusters)



- 여러 대의 전체 컴퓨터들(whole computers)이 상호 연결되어 협력하며 하나의 컴퓨터로서 동작하는 통합 컴퓨팅 자원이다.
- 클러스터를 구성하는 전체 컴퓨터는 클러스터에서 분리되면 독립적으로 동작할 수 있는 컴퓨터 시스템을 말한다.
 - 클러스터에서는 이러한 각 컴퓨터를 노드(node)라고도 한다.
- 대용량의 수치 연산에서 좀 더 고품질의 결과를 얻고 그리고 빠른 결과를 얻기 위해서는 대규모 계산이나 데이터를 처리해야 한다.
 - 기존에는 워크스테이션이나 슈퍼 컴퓨터들이 활용되어 왔으나 단일 워크스테이션으로는 충분한 성능을 제공받기에 한계가 있다.
 - 슈퍼 컴퓨터는 구축비용이 매우 높아 일반적인 연구 환경에서 사용하기에는 어려움이 따른다.
 - 결과적으로 클러스터는 이를 해결하기 위한 좋은 선택방안이 될 수 있다.

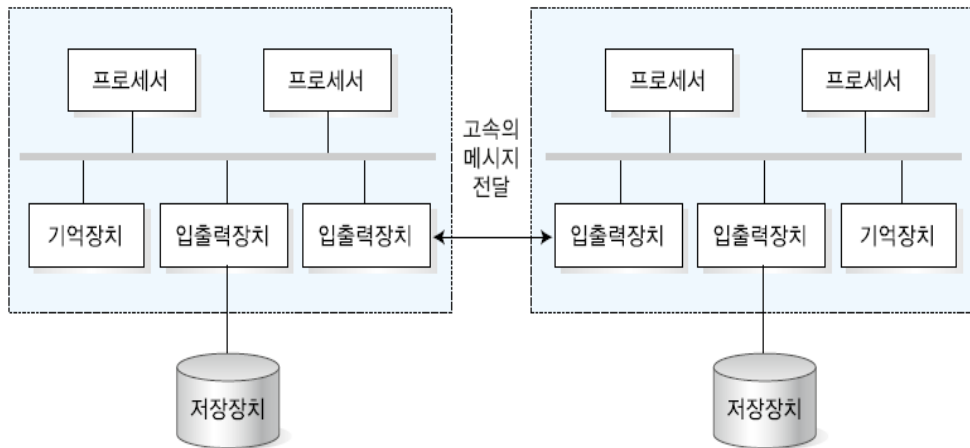


- 절대적 선형 확장성(Absolute Scalability)
 - 제일 큰 독립적 시스템보다 훨씬 더 큰 클러스터의 구성도 가능하다.
- 점진적 선형 확장성(Incremental Scalability)
 - 새로운 시스템을 점차적으로 추가하면 성능 향상이 가능하다.
- 높은 가용성(High Availability)
 - 각 노드는 독립적인 컴퓨터이므로, 독립 컴퓨터의 결함에도 서비스를 계속 제공할 수 있다. 즉, 높은 가용성을 갖는다.
- 월등한 가격/성능(Superior price/performance)
 - 클러스터는 독립적 컴퓨터의 집합으로, 고가의 대형시스템보다 더 높은 성능을 가지지만 더 낮은 비용으로 구성 가능하다

클러스터의 분류-디스크가 공유 되지 않는 구조



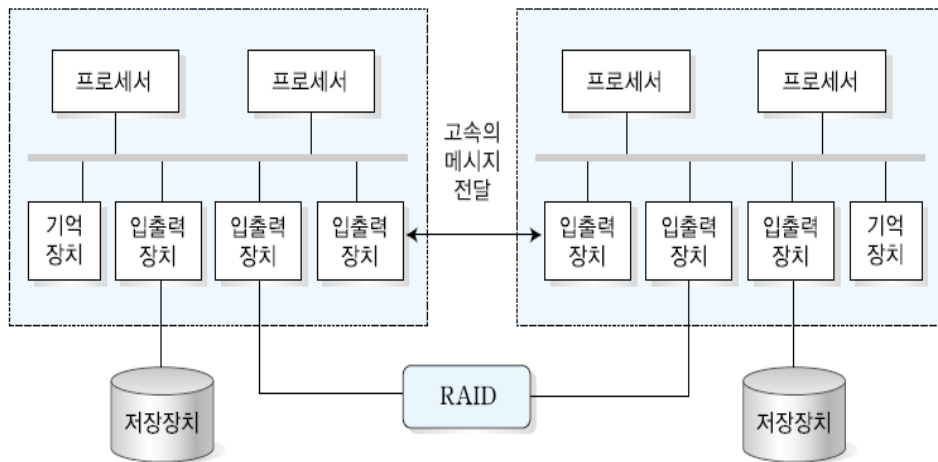
- 전체 컴퓨터들이 디스크를 공유하지 않는 클러스터 방법
 - 전체 컴퓨터 간의 연결은 LAN(Local Area Network) 또는 전용 연결망으로 연결된다.
 - 전체 컴퓨터들 간의 통신은 고속의 메시지의 전달로 이루어진다.



클러스터의 분류-디스크가 공유되는 구조



- 전체 컴퓨터가 디스크를 공유, 공유 디스크 시스템은 RAID를 사용
 - 데이터는 배열을 이루고 있는 디스크 드라이브들에 분산 저장
 - 여분의 디스크는 디스크 오류 발생시, 데이터 복구를 보장하기 위한 패리티 정보를 저장
- 디스크를 공유하는 클러스터의 구성
 - 기본적으로 전체 컴퓨터들 간에는 고속의 메시지로 연결된다. 그리고 RAID가 전체 컴퓨터 사이에서 공유된다.





● 결함관리(failure management) 기능

- 발생할 수 있는 결함들에 대한 관리를 수행하면, 모든 자원들에 서비스를 제공할 확률을 높아진다.
- 운영체제에서 지원하는 결함관리 기능은 모든 자원들이 효과적으로 운영이 되어서 고가용성 클러스터링(highly available clustering)을 만족
- 결함-허용 클러스터링(fault-tolerant clustering)
 - 모든 자원이 항상 사용 가능하도록 보장
 - 고장이 발생한 시스템에서 프로그램과 데이터 자원들을 클러스터 내 다른 시스템으로 전환하는 기능을 제공할 수 있다.

● 부하 균등(load balancing) 기능

- 사용 가능한 컴퓨터 간에 부하를 균등하게 분할해야 성능을 높일 수 있다.
- 운영체제는 부하를 균등하게 분할해 주는 기능을 지원해 주어야 한다.

수고 많이 했습니다.

