DISTRIBUTED SHARED MEMORY

Abstraction and its advantages

Distributed Shared Memory is a resource management component of a distributed operating system that implements the shared memory model in distributed systems, which have no physically shared memory. The shared memory model provides a virtual address space that is shared among all computers in a distributed system.

- It is an abstraction provided to the programmer of a distributed system.
- It gives the impression of a single memory. Programmers access the data across the network using only read and write primitives.
- Programmers do not have to deal with send and receive communication primitives and the ensuing complexity of dealing explicitly with synchronization and consistency in the message passing model.
- A part of each computer's memory is earmarked for shared space, and the remainder is private memory.
- To provide programmers with the illusion of a single shared address space, a memory mapping management layer is required to manage the shared virtual memory space.

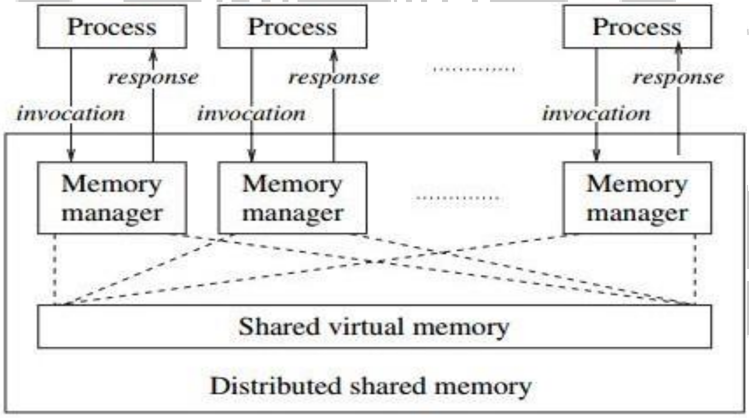

Fig : Abstract view of Distributed Shared MemoryAdvantages of DSM

- Communication across the network is achieved by the read/write abstraction that simplifies the task of programmers.

- A single address space is provided, thereby providing the possibility of avoiding data movement across multiple address spaces, and simplifying passing-by- reference and passing complex data structures containing pointers.
- If a block of data needs to be moved, the system can exploit locality of reference to reduce the communication overhead.
- DSM is economical than using dedicated multiprocessor systems, because it uses simpler software interfaces and off-the-shelf hardware.
- There is no bottleneck presented by a single memory access bus.
- DSM effectively provides a large (virtual) main memory.
- DSM provides portability of programs written using DSM. This portability arises due to a common DSM programming interface, which is independent of the operating system and other low-level system characteristics
- When multiple processors wish to access the same data object, a decision about how tohandle concurrent accesses needs to be made. If concurrent access is permitted by different processors to different replicas, the problem of replica consistency needs to be addressed.

Challenges in implementing replica coherency in DSM systems

1. Programmers are aware of the availability of replica consistency models and from coding their distributed applications according to the semantics of these models.
2. As DSM is implemented as asynchronous message passing, it faces the overhead of asynchronous synchronization.
3. Since the control is given to memory management, the programmers lose the ability to use their own message-passing solutions for accessing shared objects.

Issues in designing a DSM system:

- Determining the semantics to allow for concurrent access to shared objects.

- Determining the best way to implement the semantics of concurrent access to shared data either to use read or write replication.
- Selecting the locations for replication to optimize efficiency from the system's viewpoint.
- Determining the location of remote data that the application needs to access, if full replication is not used.
- Reducing communication delays and the number of messages that are involved under the covers while implementing the semantics of concurrent access to shareddata.