

CASE STUDY NO : 11

Date	/ /20
Page No.	55

Case Study On :- Study of object base and page base distributed shared memory.

Object - based DSM -

Object-based DSM distributed shared memory system allocates processes on different machines to communicate through passive shared objects. This paper describes the implementation of such a system on a transputer grid. The system automatically takes care of placement and replication of objects. The main difficulty in implementing shared objects is updating replicated objects in a consistent way. We use totally-ordered group communication for this purpose. We give four different algorithms for ordering broadcasts on a grid and study their performance. We also describe a portable runtime system for shared objects.

The page-based DSM systems that we studied use the CPU hardware to trap accesses to missing pages. While this approach has some advantages, it also has some disadvantages. In particular, in many programming languages data are organized into objects, packages, modules or other data structures each of which has an existence independent of the others. If a process references part of an object, in many cases the entire object will be needed. So it makes sense to transport data over the network in units of objects not in units of page.

The shared-variable approach, as taken by Munin and Midway is a step in the direction of organizing the shared memory in a more structured way, but it is only a first step. In both systems the programmer must apply information about which variables are shared and which are not, and must also provide protocol information in Munin and association information in Midway. Errors in these annotations can have serious consequences.

Objects -

An object is a programmer-defined encapsulated data structure as depicted. It consists of internal data, the object state, and procedures, called methods or operations that operate on the object state. To access or operate on the internal state, or something else, direct access to the internal state is not allowed. This property, called information hiding. Forcing all references to an object data to go through the methods helps structure the program in a modular way.

In an object-based distributed shared memory, processes on multiple machines share an abstract space filled with shared objects, the location and management of the objects is handled automatically by the runtime system. This model is an contrast to page-based DSM systems such as TVM, which just provide a raw linear memory.

of bytes from 0 to some maximum.

Page - Based DSC -

In a page based DSC, the data to be shared among processes are organized as logical fixed-size pages that are distributed over multicomputers. Whenever a page required by a process is locally available, the DSC grants access to the required page via the memory management unit (MMU). This requires simple memory access to the secondary storage that does not involve network access on the other hand, when a process tries to access a page that is non-local to the machines, then the page faults to the DSC. The DSC software should now search for the page on a network of computers in a true distributed system and when found is fetched from the source computer. The faulting instructions is then restarted and can now complete.

Page based or block based DSCs are an extension of traditional virtual memory systems thus are usually implemented at the hardware and/or OS layers. Because the implementations is at the HW and/or OS layers there is complete transparency with respect to memory system, i.e. memory system is completely hidden from users of the system. It is difficult to choose the right size for the page and/or block because page and/or block not only depends on the system characteristics but also on application.

Advantages -

- 1) Any Dscl can be used, since the run-time will decide the fate of objects for their transaction from one processor node to another. This will result in the flexibility offered to users.
- 2) Any consistency model as that in the oDscl can be used.
- 3) Object access is secured since only the member method of the object can operate on them.

Disadvantage -

- 1) Integration of existing system has huge overhead although changes need to be done only in the oDscl
- 2) Integrated software Dscl like this would have huge memory requirements.
- 3) Complexity of the system can never be ignored.

Conclusion -

Many existing PDscls and ODscls are present in the real world with different features. All these systems have their pros and cons. As mentioned in the earlier section, ODscls have scope for a lot of improvements.

