A FRAME WORK FOR RELIABLE DISTRIBUTED COMPUTING

M.UVA RANI M.E.,
HEAD & ASSISTANT PROFESSOR IN B.SC(IT),
S.B.K.COLLEGE,ARUPPUKOTTAI.

Abstract : Most application level fault tolerance schemes in literature are non-adaptive in the sense that fault tolerance schemes incorporated in applications are usually designed without incorporating information from system environments such as amount of available memory and the local network I/0 bandwidth. Fault tolerance will decide whether the current process to be continued or redo based on the number of failure systems in the DCE. if failure system is less then total number of participants then failure recovery can be done. Recovery can be done with the help of system properties stored in that failed system or by the adjacent system.

I. Introduction:

Overview of Distributed System:

A distributed system is a collection of loosely coupled processors interconnected by a communication network. From the point of view of a specific processor in a distributed system, the rest of processors and their respective resources are remote its own resource are local.

A distributed operating system provides users with access to the various resources that the system provides.

II. Connection Establishment:

The server and the clients establish the connection by the TCP/IP socket. It allocates a port number to the client to establish the UDP connection. By this Port number it communicates with the client and it increments port number to allocate for next client.

Brief Description of Connection Establishment:

Input : IP Address of server.

Output : Connection will be established and clients screen will be opened.

III. Job Submission And Distribution:

The client selects a job from its file and it sends it to the server by the data send class. The server receives the job file and it splits it by using the job split method. It splits by the number of clients in online. It sends the split jobs to the client

Brief Description of job submission and distribution:

Input : File selected from client.

Output: The splitted jobs send to the clients.

IV. Job Processing and combination

The client receive the job from the clients to be processed. It Process the job and sends the result to the server by data send class. The received results are combined in the server by the combine method. If any process is failed while processing the server sends the rest job which is not processed by it to the client which is finished soon. The client sends it also to the server and all are combined and result is send to the client which submitted the job. The processed job is stored as a file in the client.

Brief Description of Processing and Combination:

Input: The job send from the server.

Output: The Processed job is sent to client and it is stored in a file.

V. Fault Tolerance and Failure Recovery:

When the client is processing the job, if any fault occurs like system crash, link failure, the server finds the fault through snapshot signal and it recovers it. The recovery is done by the client which finishes the job. The server gives the unfinished job to the client and it finishes it and returns to the server.

Brief Description of Fault Tolerance and Failure Recovery

Algorithm: Snapshot Algorithm

Input: The failed job.

Output: The recovered job.

VI. Snapshots Algorithm:

An algorithm whose task is to analyze properties of computation , usually from another algorithm. An Important building block in the design of algorithms operating on system computation is a procedure for computing and storing a signal configuration of this computation called snapshot. Snapshot is applicable in three places.

First ,Properties of the computation as far as they are reflected within a single configuration, can be analyzed Off-Line ,i.e., by an algorithm that inspects the(fixed)snapshot rather than (varying) actual process states. These properties include stable properties; a property P of configurations is stable if p(r) r->=>p(d).if a computation ever reaches a configuration d from then on. Consequently , if P is found to be true for snapshot of the configuration, the truth of P can be concluded for all configuration from then on.

Second ,a snapshot can be used instead of the initial configuration if the computation must be restarted due to process failure. To this end the Local state Cp for process P, captured in the snapshot ,is restored in that process, after which the operation of the algorithm is continued.

Third, snapshots are a useful tool in debugging distributed programs. An Off-Line analysis of a configuration taken from an erroneous execution may revel why a program does not act as expected.

VII. Conclusions:

Mainframes and Parallel computers are highly reliable and cost number of crunches

The recent decade of client server technology evolution indicates cost as a prime factor

There by, identifying distributed object based solution Linux clusters.

With full advantage of economy distributed object based solution still stays as a bumbling amateur.

As the first pedestal, we had designed a Reliable Distributed Computing Environment in Windows Platform.

The paper was listed with a cry problem of Encryption.

It was successfully and has paved hope towards a lot of future scope.

VIII. References:

[1].Zizhong chen,Ming Yang,Jack Dongarra.Self adaptive Application Level Fault Tolerance for Parallel and Distributed Computing IEEE Paper 2007.

[2]. Andrew s.Tenenbaum-"Distributed Operating System"

[3].Bruce Schneier-"Applied Cryptography "WSE Wiley- 2nd Edition.

[4].Nancy A.Lynch"Distributed Algorithm"

[5].Patrick Naughton,Herbert Schildt-"The Complete Reference Java 2"

[6].Don Box"Essential COM"Addition Wesley.

[7].Jeremy Rosenberger "Tech Yourself CORBA"Sams Publishing Tech media.

[8].Bill McCarty and Luke Cassady- Dorion "Java Distributed Objects" sams tech media I Edition.

Website

http://www.iids.org

http://www.distributed.net

http://www.opengroup.org

http://www.vbip.com/winsock/distributed-algorihm