Fault Tolerance Under Unix

Anita Borg
Wolfgang Blau
Wolfgang Graetsch
Ferdinand Herrmann
Wolfgang Oberle

Presented by Marina Surlevich

1

Introduction to Targon/32 System

- Hardware and software based system
- Allows recovery of arbitrary programs
- Recovers from both hardware and software failure
- All processors available for productive execution in the absence of failure
- Degree of back up is set by the user
- Backup and recovery transparent to the user

2

Introduction to Targon/32 System

- Backup and recovery criteria
  - A crashed process's state must be available
  - All messages that would have been available to the primary at that state or since that state was reached must be available in the correct order
  - Process must behave deterministically

3

System Architecture

- LAN of 2 to 16 machines connected via a fast dual bus
  - three processors on each machine
- Peripheral devices are dual-ported
- Root pair
  - Two machines connected to disk containing the root file system
- Atomic three-way message delivery
- Messages arrive in order

4

Atomic Three-Way Message Delivery

- All machines listen for their address to come across the bus
- The sender requests bus mastership. On receipt of mastership, transmits the three destination machine identifiers and waits
- A machine seeing its address on the bus prepares to receive. If cannot receive a message, sends a NACK.
- A machine that can neither receive not NACK is dead
- If the sender receives no NACK within a specified period of time, it sends the message across the bus once
- The message is picked off the bus by each of the ready receivers

5

Process

- Fundamental recoverable unit of execution
- Controlled by OS kernel
- Communicate with each other and receive all input via messages

6

Processes and Their Backups

- Primary process has an inactive backup process on another machine
- All messages to primary are broadcasted to backup (three-way atomic broadcast)
- All messages sent by primary are counted by backup as “writes-since-sync”

7

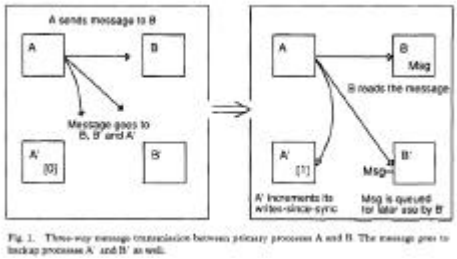


Fig. 1. Three-way message transmission between primary processes A and B. The message goes to backup processes A' and B' as well.

8

Processes and Their Backups

- Primary and backup are periodically synchronized (Sync operation)
  - System defined number of messages or
  - Amount of time since last synchronization
- On failure of primary backup will
  - Demand-page in address space of primary since last synchronization
  - Recompute current state

9

Server Processes

- Execute in their own address space
- Provide services to other processes
- Modularity, modifiability, distribution, recoverability
- Servers that always exist: file servers, page servers, TTY servers, raw servers, process server
- Page servers and root file server reside permanently in memory

10

Process Families and Backup Modes

- Processes divided into families
- All members reside on one machine and have common ancestor
- Server processes – single member families
- Process family – basic backed-up unit
- Created by wexec (walking exec) – has an argument to specify how to backup
  - Quarterbacks
  - Halfbacks
  - Fullbacks (not implemented)

11

Interprocess Communication

- Channel – recoverable two-way communication mechanism
  - Example: open file is represented by a channel to the file server managing the file
  - Explicitly used or implicitly used
- Messages are placed on general input queue
  - Arrival number assigned to each message
  - Processed in arrival order

12

Backup and Synchronization of User Processes

- Creation of backup processes
  - When process created “birth notice” is sent to parent’s backup
  - When parent synchronizes, children are forced to sync

13

Synchronization

- Sync – automatically called by the kernel
- Dirty pages sent to the page server
- Sync message
  - Sent to process’s backup, to page server and it’s backup
  - Contains
    - Machine-independent information about process’s state
    - Channel information for every open channel
    - Information to allow construction of kernel stack on recovery

14

Deterministic Execution

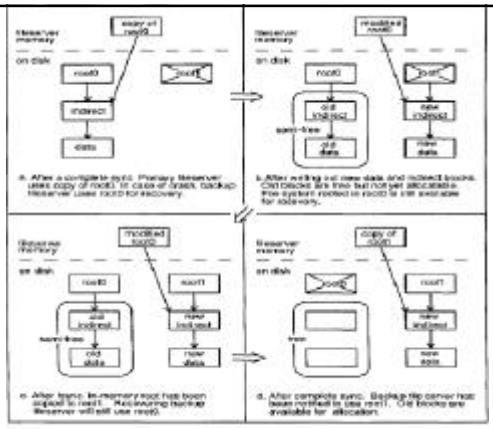
- Process and its backup must be insulated from local differences
- Synchronous interaction
  - Information returned by kernel is maintained
  - Page faults are transparent
- Asynchronous interaction
  - Primary syncs before handling any signal
  - Ignored signals are counted and removed from primary’s queue

15

Backup and Synchronization of Servers (File Server)

- Behaves differently from user processes
- Modified srv_read and srv_write
  - History array of the form <u2, d2, d1, u1, d3>
- Sync and fsync are combined (clears history array)
- Recovers explicitly

16



Crash Detection and Handling

- Machines organized in a virtual ring
- Each machine expects regular report from left neighbor
- On failed communication
  - Determine if it can communicate with another machine
    - If not, it assumes it crashed
    - If yes, order the uncommunicative machine to die
  - Broadcast a “machine-dead” message
  - Locate new left neighbor

18

Process Recovery

- For primary processes whose backups were lost
  - Primary Quarterbacks marked “not backed up”
  - Primary Halfbacks marked “not currently backed up”
- Kernel must do:
  - Allocate and initialize structures for local kernel state and memory mapping
  - Request a list of pages from page server
  - Set up kernel stack from latest sync information
  - Put process on the run queue

19

Roll Forward

- Period during which a process reexecutes code that was previously executed by primary
- Messages are not sent while “writes-since-sync” count is positive.
- Before forking, birth notices are checked
  - Birth notice exists
    - Primary child was synced (no new process is created)
    - Primary child was not synced (child process is created)
- Process not allowed to sync until end of roll forward.

20

Machine Reintegration and Rebackup

- When kernel is booted, machine sends “machine-up” message to process server
- New backup processes created for halfbacks that lost their primaries or backups
- Resync is done for all families that have to be rebacked up
- Switch sync can be done to reverse the roles of primary and backup

21

Machine Reintegration and Rebackup (Resync)

- Reconstruct either a wexec message or a birht notice and send it to backup machine
- Force all children to resync
- Sync
- If last in the family to finish syncing, send “notify” message for local and backup machines
- Send the backup machine copies of all messages currently linked to the process’s routing table queues that arrived before the notify message

22

Performance

- Affected by:

- Distributed message-based architecture
- Overhead for fault tolerance

23

Overall Performance

- Distributed message-based system organization reduces performance by 15%
- Each additional machine increases the overall capacity by 70%
- Fault tolerance reduces performance by 10%
- If benchmarks are run on a two-machine system, the performance increases 1.6 times.
- The delay experienced by the user whose primary process dies is 5~15 seconds.

24

Conclusion

- System can recover from a large class of kernel software faults
- System can recover from hardware failures
- Fault-tolerant operation is automatic and transparent to the user
- Unaffected processes are not penalized during recovery

25