

METHOD AND APPARATUS FOR SPLIT-BRAIN AVOIDANCE IN A MULTI-PROCESSOR SYSTEM

This invention relates generally to fault-tolerant multi-processor systems. In particular, this invention relates to methods for improving the resilience of a multiprocessor system in partial and total communication failure scenarios.

RELATED PATENT APPLICATIONS

U.S. patent application Ser. No. 08/265,585 entitled, "Method and Apparatus for Fault-Tolerant Multi-processing System Recovery from Power Failure or Drop-Outs," filed Jun. 23, 1994, naming as inventors Robert L. Jardine, Richard M. Collins and Larry D. Reeves, under an obligation of assignment to the assignee of this invention, with Attorney Docket No. 010577-031900/TA 271;

U.S. Pat. No. 5,687,308, issued Nov. 11, 1997, entitled, "A Method to Improve Tolerance of Non-Homogeneous Power Outages," filed Jun. 7, 1995, naming as inventors Robert L. Jardine, Richard M. Collins and A. Richard Zacher, under an obligation of assignment to the assignee of this invention;

U.S. patent application Ser. No. 08/790,030 entitled, "Method and Apparatus for Node Pruning a Multi-Processor System for Maximal, Full Connection During Recovery," filed on the same date as the instant application, naming as inventors Murali Basavaiah and Karoor S. Krishnakumar, under an obligation of assignment to the assignee of this invention, with Attorney Docket No. 010577-040000/TA 333 DIV 1;

U.S. patent application Ser. No. 08/790,268 entitled, "Method and Apparatus for Toleration of Lost Timer Ticks During Recovery of a Multi-Processor System," filed on the same date as the instant application, naming as inventors Murali Basavaiah, Karoor S. Krishnakumar and Srinivasa D. Murthy, under an obligation of assignment to the assignee of this invention, with Attorney Docket No. 010577-039900/TA 333 DIV 2; and

U.S. patent application Ser. No. 08/789,257 entitled, "Method and Apparatus for Distributed Agreement on Processor Membership in a Multi-Processor System During Recovery," filed on the same date as the instant application, naming as inventors Robert L. Jardine, Murali Basavaiah, Karoor S. Krishnakumar and Srinivasa D. Murthy, under an obligation of assignment to the assignee of this invention, with Attorney Docket No. 010577-039800/TA 333 DIV 3.

BACKGROUND OF THE INVENTION

Distributed, shared-nothing multi-processor architectures and fault-tolerant software using process pairs require that all processors in a system have a consistent image of the processors making up the system. (The NONSTOP® KERNEL operating system (NONSTOP® is a registered trademark and NONSTOP® KERNEL is a trademark of Tandem Computers Incorporated), available from the assignee of this application is an example of such fault-tolerant software.) This consistent system image is crucial for maintaining global system tables required for system operation and for preventing data corruption caused by, say, an input/output process pair (IOP) of primary and backup processes on different processors accessing the same I/O device through dual-ported I/O controllers or a shared bus (such as SCSI).

Detection of processor failures occurs quickly with an IamAlive message scheme. Each processor periodically

sends IamAlive packets to each of the other processors in the system. Each processor in a system determines whether another processor is operational by timing packets from it. When the time interval passes without receipt of a packet from a given processor, the first processor decides that the second might have failed.

In older systems, before regrouping was implemented, the following could occur when the second processor then sent a packet to the first. The first processor judged the second to be functioning improperly and responded with a poison packet. The first processor ignored the content of the packet from the second.

Ultimately, many or all of the other processors could end up ignoring the affected processor (except to try to stop it). The affected processor was, in effect, outside of the system and functioning as if it were an independent system. This condition was sometimes called the split-brain problem.

Without regrouping, the following situations can occur: Both of the processes in a process pair running on different processors can regard themselves as the primary, destroying the ability to perform backup functions and possibly corrupting files. All system processors can become trapped in infinite loops, contending for common resources. System tables can become corrupted.

Regrouping supplements the IamAlive/poison packet method. Regrouping uses a voting algorithm to determine the true state of each processor in the system. Each processor volunteers its record of the state of all other processors, compares its record with records from other processors and updates its record accordingly. When the voting is complete, all processors have the same record of the system's state. The processors will have coordinated among themselves to reintegrate functional but previously isolated processors and to correctly identify and isolate nonfunctional processors.

Regrouping works only when physical communication among processors remains possible, regardless of the logical state of the processors. If a processor loses all of its communications paths with other processors, that processor cannot be regrouped. It remains isolated until communications are restored and the system is cold loaded. (Such a processor usually stops itself because its self-checking code cannot send and receive message system packets to and from itself.)

A processor's logical state and its condition are distinguished. A processor has two logical states in a properly configured system: up or down. However, a processor has three conditions: dead, which is the same as the down logical state; healthy, which is the same as the up logical state; and malatose, which is described further below.

A processor is dead if it does not communicate with the rest of the system. Dead processors include those, for example, that execute a HALT or a system freeze instruction, that encounter low-level self-check errors such as internal register parity errors, that execute infinite loops with all interrupts disabled, that execute non-terminating instructions due to data corruption or that are in a reset state.

Dead processors are harmless, but the regrouping algorithm removes them from the system configuration. Other processors detect dead processors and declare them down.

A processor is healthy if it is running its operating system (preferably, the NONSTOP® KERNEL operating system available from the assignee of the instant application) and can exchange packets with other processors (preferably, over a redundant high-speed bus or switching fabric) within a reasonable time. The regrouping algorithm prevents a processor declaring down a healthy processor.