the Availability Digest

The Sombers Group™

The coming day of fault tolerance

Dr. W. H. Highleyman

Reprinted from
THE NEWSWEEKLY FOR THE COMPUTER COMMUNITY
COMPUTERWORLD
SEPTEMBER 10, 1984 VOL. XVIII, NO. 37

© 2006 Sombers Associates, Inc., and W. H. Highleyman
www.availabilitydigest.com

The coming day of fault tolerance

By Wilbur H. Highleyman

A little more than four years ago, my In Depth series, "Survivable Systems" [CW, Feb. 4-Feb. 25, 1980], appeared in *Computerworld*. At that time, Tandem Computers, Inc. offered the only commercially available fault-tolerant system, the Nonstop. Now that there are about two dozen offerings from domestic and foreign manufacturers, it is clear that fault tolerance has been accepted as a concept with a future.

As with memory and languages and operating systems, fault tolerance will become a subconscious requirement. We just wouldn't think of building a system without it.

It is my guess that within the next 10 years fault tolerance will be as common as higher level languages, multiuser operating systems and megabyte memories.

I first became involved in fault-tolerant systems in the mid-1960s, when my company at that time, Data Trends, Inc., built a triplexed totalizator system for the New York Racing Association. The system was designed for the Aqueduct, Belmont and Saratoga racetracks and used Honeywell, Inc. H200s. Remember them? They were 6-bit machines designed for commercial data processing.

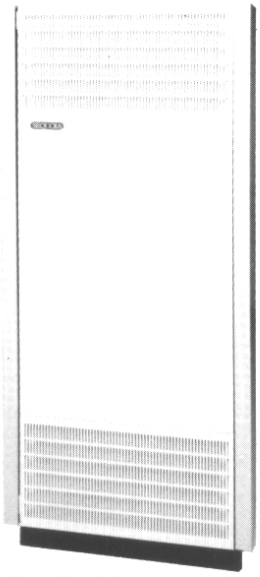
The system was quickly followed by a network of duplexed Digital Equipment Corp. PDP-15s acting as the telex switches still used by ITT World Communications, Inc. The floor of the Chicago Board of Trade was first automated with triplexed DEC PDP-8s.

The Sombers Group was later involved with other systems. One included telex switches for the British Post Office using General Automation, Inc.'s SPC-16. Another, the data base manager for the (New York) *Daily News* editorial system, used duplexed Prime Computer, Inc. 200s.

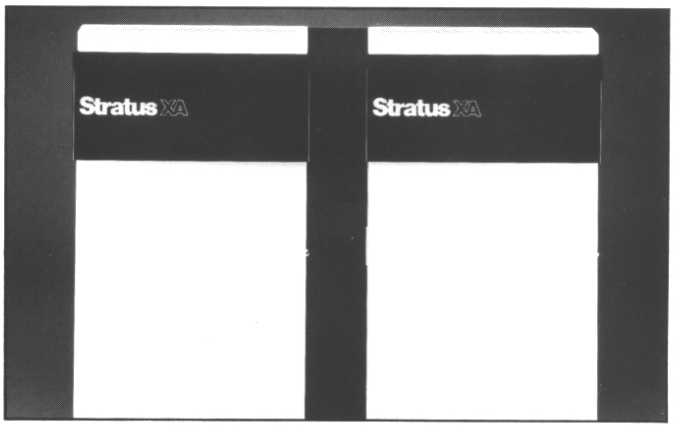
All of these fault-tolerant systems were pre-1976, and all had the following characteristics:

- Each used a homegrown operating system supporting a fault-tolerant strategy that was quite specific to the application.
- Each ran as a single processor with a shadow standby that could take over without losing a transaction (or a call in progress). Triplexed systems simply provided a third cold standby.
- Memory sizes were small by today's standards (16K bytes to 64K bytes).
- All programming was done in assembly language.

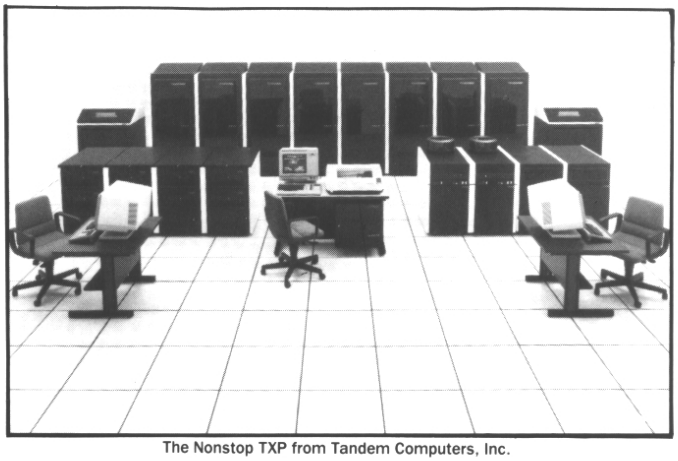
In those days, we built the systems from scratch and squeezed the last bit of performance



Sequoia's fault-tolerant machine, to be introduced Sept. 18



The Stratus strategy: hardware redundancy



The Nonstop TXP from Tandem Computers, Inc.

© 2006 Sombers Associates, Inc., and W. H. Highleyman
www.availabilitydigest.com

out of the hardware. After all, hardware was expensive, and software was cheap (at least, that was management's perception).

Then came 1976 and Tandem, and our little world of custom fault-tolerant systems turned upside down. Tandem had the daring to introduce a generic fault-tolerant product — one that was applicable to almost any application. Just learn to checkpoint, Tandem said, and you can build any application to run fault tolerant.

Tandem was right. The custom fault-tolerant business of The Sombers Group vanished almost instantly, to be replaced with experience on dozens of new Tandem-based systems. Of course, checkpointing did not turn out to be the trivial exercise touted by Tandem. And there were failure windows not even anticipated. But with ever-increasing experience and continuing improvements in Tandem products, the building of these systems has become more straightforward.

Today, these systems can be found in almost any application, except those which require ultra-fast response time or are products aimed at very low-cost markets (these are still candidates for custom operating systems).

The most amazing part of the Tandem story is that the company held this market captive for six or seven years — unheard of in today's technology. It was not until 1982 that Stratus Computer, Inc. became a serious competitor. Synapse Computer Corp. emerged next, followed by dozens of others. Now, within just two short years, the field is full with fledgling hopefuls. Each brings a new idea to the fault-tolerant marketplace — a new piece to the puzzle — begging the question of where is all this going.

What is fault tolerance?

As a general statement, a fault-tolerant system is one in which any failure is transparent to the user. However, this general statement is unusually demanding. For one thing, it implies that any sequence of failures will be user-transparent. The reliability of today's hardware is such that tolerance of any single failure is sufficient, given sufficiently rapid repair time.

The statement also makes no allowance for reduced performance in the (presumably) rare event of a failure. This is often tolerable, as long as all functions are still available to the user community. We will therefore use a relaxed definition for purposes of the following discussion: A fault-tolerant system is one in which any single failure is functionally transparent to the user.

In order to be fault tolerant, a system must have the following characteristics:

Redundancy. Each critical component must be replicated so that it can be replaced upon failure.

Fault detection. The system must be able to identify automatically a component that has failed.

Isolation. The failed component must be isolated, electrically and logically, from the rest of the system.

Reconfiguration. The system must reconfigure itself to continue uninterrupted operation in the presence of this fault.

Repair. The faulty component must be repairable without affecting normal ongoing system operation.

Recovery. Once repaired, the failed component must be reintroduced into the system.

Data base integrity. No failure mode shall be allowed to contaminate the data base.

What's happening now?

Tandem, located in Cupertino, Calif., is the granddaddy of generic fault-tolerant systems. Tandem introduced a multicomputer system in which up to 16 independent computers were interconnected by a dual high-speed bus. Within this structure, fault tolerance was implemented via software. Each operating program (or process) running in one computer could have a back-up process running in another computer.

Via software-implemented checkpoints at critical points in the process, the back-up process is informed of the primary process' exact state. Thus, if the primary process should fail (due to a processor failure, for example) the back-up process could take over very quickly (within seconds) from the last known state. Done properly, the user is unaware of the failure.

Given that Tandem's approach reflected the technology of the mid-1970s, it was a very advanced concept and obviously is holding its own today. (Tandem continues to command the bulk of the market for fault-tolerant systems.) However, this "software" approach has certain drawbacks. Checkpointing imposes a nontrivial system overhead, and the understanding and generation of solid checkpointing strategies is a little-understood art.

Using the microprocessor technology of the '80s, Stratus Computer of Natick, Mass., introduced a totally new idea — fault tolerance strictly through hardware redundancy. Taking the Stratus processor as an example, it is really four microprocessors (Stratus uses the Motorola, Inc. 68000) arranged as two pairs of duplexed processors. All four microprocessors are run in lock step at the system clock frequency. The outputs of each microprocessor within a duplexed pair are continually checked for consistency.

As long as all outputs are the same, both pairs operate in parallel. If one duplex pair should find an inconsistency, it immediately disconnects itself from the system bus, but the other duplex pair continues operations.

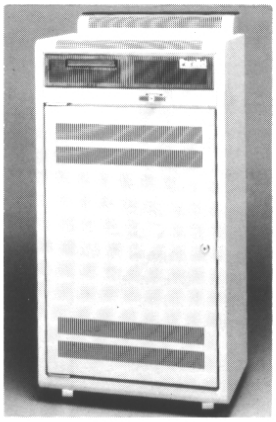
Along with its equally secure memory, disk and communications units, a Stratus "processing module" will continue uninterrupted processing in the event of any single failure. Stratus allows up to 32 processing modules to be connected via a high-speed bus, although this capability is used solely for expansion and not recoverability as in Tandem's case. (Note: To be more accurate, each Stratus processor really is made up of a multiple of four microprocessors, since there is one set for executive work and one or more sets — depending on the model — for application work. Each set is quadraplexed and runs in lockstep.)

Hardware approach

Stratus solved the overhead problems of Tandem's software approach through its "hardware" approach of quadraplexing. On the surface, this appears to be a more expensive, though higher performance, solution. However, it is not clear from actual field experience that either is true: Both systems appear to be price/performance competitive.

Then enter Synapse Computer Corp. of Milpitas, Calif. Synapse argues that the multicomputer approach of both Tandem and Stratus is inefficient for two reasons. One is that many applications require several cooperating processes, especially in transaction processing. Terminal requestors communicate with transaction servers, which communicate with a data base manager.

In loosely coupled systems such as these, the interprocess communications mechanism is lengthy (typically a few milliseconds) and represents a significant portion of the system load. Secondly, though the load of the system can be shared among the computers, it cannot be easily dynamically balanced, since processes must be preassigned to a particular computer. It is not uncommon for one computer to be running at 90% to 100% load while others are running at 10% to 50% load.



The Parallel 300

Synapse introduced a closely coupled multiprocessor system (as opposed to a multicomputer system) in which up to 28 general-purpose or I/0 processors (again, the Motorola 68000) use one common memory. This improves efficiency in two ways. Interprocess communications is now via direct memory and is measured in microseconds rather than milliseconds.

Furthermore, all general-purpose processors can work off a common task queue, thus keeping all equally busy. A process is no longer assigned to a single processor; on each dispatch it is assigned to the next free processor.

Contaminate memory

Of course, a closely coupled architecture has a horrible fault-mode because any sick processor, through certain hardware failures, can contaminate main common memory. And the failure of a main memory module also represents a catastrophic system failure. Synapse, therefore, addresses fault tolerance with a somewhat different philosophy. The company submits that, in many applications, it is perfectly reasonable to require users to reenter their current transaction in the event of a failure — provided that failures are infrequent and that the integrity of the data base is maintained.

To this end, Synapse maintains sophisticated logs of all disk activity. In the event of a failure of any kind, the system is brought down, automatically reconfigured, and any transactions in progress are rolled back. It may take a few moments a few times a year, but Synapse argues that the resulting price/performance of its "transaction" approach to fault tolerance far outweighs this inconvenience.

A common thread of all of these systems is expandability and transparency. Applications can be written and systems installed without worrying about outgrowing the size of the box. If volumes increase, functions proliferate, or (heaven forbid) the application was underestimated, simply add modules — more processors, more disks, more terminals, whatever is needed. And if you fill up a system or need to distribute it, interconnect multiple systems with anything from local-area networks to satellite links.

The beauty of this expandability is its transparency to the applications programs. The operating system knows the name and whereabouts of every processor, file and peripheral in the system. As these are moved around in an expanding system, the applications programs need know nothing of this configuration change and, therefore, require no modification. A few minutes of modifying the system generation parameters does it all.

Service is another area of innovation. Tolerance of a single failure is only of value if that fault can be rapidly repaired. Led by Stratus Computer and followed by some of the newer entries, a failed system will automatically place a call to a centralized service facility. The fault can be diagnosed over the dialed data link, and a replacement component is sent by courier overnight to the customer for replacement. Often, the customer will receive the replacement component before he has even noticed the system alarm designating a failure. I suppose all that is left is training the courier service to also replace the part.

Federal DPers will meet on fault tolerance

Wilbur H. Highleyman will speak on state-of-the-art fault tolerance at the Federal Computer Conference, to be held Sept. 18-20 at the Washington, D.C. Convention Center.

This seventh annual event will include sessions, workshops and professional enhancement seminars.

More information is available from the Federal Computer Conference, 286 Boston Post Road, Wayland, Mass. 01778.

© 2006 Sombers Associates, Inc., and W. H. Highleyman
www.availabilitydigest.com

The Unix-based Auragen System 4000

Of course, a closely coupled architecture has a horrible fault-mode because any sick processor, through certain hardware failures, can contaminate main common memory.

Three philosophies

We now have three distinct fault-tolerant philosophies introduced by the first three entries (at least, domestically) into the generic fault-tolerant field. There is the software approach by Tandem, the hardware approach by Stratus and the transaction approach by Synapse.

Each approach has its pluses and minuses, perhaps most easily expressed by the minuses. The software approach imposes system overhead, the hardware approach imposes hardware overhead, and the transaction approach imposes recovery time overhead. It is interesting that the newcomers to the field following Tandem, Stratus and Synapse use one or a combination of these techniques.

Auragen Systems Corp. of Fort Lee, N.J., offers a multiple computer system in which up to 32 processor clusters can be interconnected via dual high-speed buses. Each processor cluster is an independent computer system which is itself a closely coupled configuration of three microprocessors (one executive and two application processors) communicating with its own common memory, files and peripherals.

Auragen's fault-tolerance approach is similar in many ways to Tandem's software approach, except that checkpointing is done automatically at the operating system level. Therefore, it requires no application awareness and is more efficient during normal operation, but with longer recovery times. Auragen's fault-tolerant system is a Unix-based system. The company claims that any Unix program will run fault tolerant with no changes.

Tolerant Systems Corp. of San Jose, Calif., has announced a system in which multiple System Building Blocks can be configured in a variety of ways. Each System Building Block contains two National Semiconductor Corp. 32032 microprocessors (one for real-time executive work and one for application tasks), memory and I/O interfaces.

In a typical configuration, System Building Blocks are configured in pairs for handling specific system functions. For instance, one pair of building blocks would act as communication servers, one pair as application servers and one pair as file servers. Normally, both System Building Blocks in a pair would share the load.

However, if one fails, all traffic is passed through the surviving building block. Tolerant's system recovers from a fault via a transaction recovery similar to Synapse. Incomplete transactions are backed out of the data base, and the user reenters those once recovery is complete.

Sequoia Systems in Marlboro, Mass., combines the hardware and transaction approaches. Its product, to be introduced Sept. 18 at the Federal Computer Conference in Washington, D.C., is a closely coupled system like that of Synapse. Up to 64 general-purpose processors and 96 I/O processors communicate with common memory over a pair of high-speed buses.

Unlike Synapse, each general-purpose processor is composed of a pair of Motorola 68000 microprocessors operating in Stratus-like lockstep. Thus, each is completely self-checking. Memory elements and I/O processors are also self-checking. Like the Synapse system, if there is an error, incomplete transactions are backed out and restarted following recovery.

Modified stance

Nohalt Computers, Inc. of Farmingdale, N.Y., takes a modified stance on fault tolerance. In its Failsafe system, up to 16 application processors communicate with a mirrored, redundant data base manager. Each application processor provides a single user environment, supporting a user terminal and up to three other peripherals associated with its application, such as a printer or a communications line. If there is a failure in the data base manager, system operation continues. A failure in an application processor will take that user out of service.

Parallel Computers of Santa Cruz, Calif., provides a system in which up to five parallel processing units can communicate over a single simplexed bus to disk controllers and communications controllers. Each parallel processing unit is actually two processors, each Motorola 68000-based. Each half of a parallel processing unit is independently processing the same transactions as the other half. The unit halves checkpoint to each other to keep each one in sync with the other's progress. If one fails, the other carries on.

The above systems represent many of the current domestic entries in the generic, expandable fault-tolerant system market (though the Parallel system is aimed more specifically at supervisory control and data acquisition applications). Other systems worth noting that are less generic and not expandable are from August Systems, Inc. and Syntrex, Inc.

August Systems, of Salem, Ore., markets a triplexed voting system to the process control and supervisory control and data acquisition marketplaces. All inputs are processed by each of three independent Intel Corp. 8086-based processors, and the results, whether digital or analog, are voted on. If there is a disagreement, two out of three determine the result. The repeated failure of one processor to agree with the other two takes it out of service.

Syntrex, Inc., of Eatontown, N.J., markets its fault-tolerant Gemini system to the word processing market. Similar in architecture to the Nohalt system, Gemini is composed of an 8086-based redundant data base manager and up to 14 8086-based intelligent Aquarius terminals. A failure in the data base manager is transparent to the user, whereas an Aquarius failure puts that user out of service.

Some boards of directors don't want to hear anything but Big Blue, and Big Blue doesn't have a fault-tolerant system. But even this argument is dissipating as granddaddy Tandem rapidly approaches becoming a billion-dollar company.

Of noticeable absence in the above discussion are the major computer manufacturers established prior to Tandem. Not one has announced a generic, expandable fault-tolerant system, though there are some entries in one form or another. Both Perkin-Elmer Corp., with its Resilient system, and AT&T, with its 3B20D, market a dual-processor fault-tolerant system. But once the user grows out of this "box," he is faced with difficult decisions on how to expand. With Perkin-Elmer, the user at least can climb the 3200 product line, from the small 3205 to the powerful 3200 MPS.

And then there is DEC, the perennial hopeful for a major entry in the field. After one false announcement and rapid withdrawal, DEC

© 2006 Sombers Associates, Inc., and W. H. Highleyman
www.availabilitydigest.com

finally announced VAX cluster, which can support up to 16 nodes interconnected by a high-speed dual bus. Each node can be a VAX (780 or 750) or a disk control unit (hierarchical storage controller). DEC has delivered VAX cluster hardware, but software is still forthcoming, even for such elemental operations as file sharing, not to mention file mirroring and fault recovery. And it appears that fault recovery will be implemented by checkpointing process context to disk — a backward step from software checkpointing?

And then there is DEC, the perennial hopeful for a major entry in the field. After one false announcement and rapid withdrawal, DEC finally announced VAX cluster, which can support up to 16 nodes interconnected by a high-speed dual bus.

Shaping the future

Some interesting common paths are beginning to take shape that may set a precedence for future offerings. One is the embracing of the software, hardware and transaction-recovery techniques introduced by Tandem, Stratus and Synapse. So far, there do not appear to be any new philosophies beyond these introduced.

Another common path is the inherent expandability of these systems. I would not expect to see any new ventures introducing systems limited to dual processors. And still another is the wealth of software tools that have come to be expected, from the normal complement of higher level languages to networking, data base management and transaction processing tools.

Development efforts appear to be focusing (and appropriately so) on the use of existing technology to reduce development effort. To wit, with respect to the newcomers:

Processor — All but Tandem use microprocessor technology. Five use Motorola 68000 (Stratus, Synapse, Auragen, Parallel and Sequoia).

Operating system — Many of these systems are Unix-based, including Auragen, Tolerant, Sequoia, Nohalt and Parallel.

System language — All use a high-level language for a system language. C is predominant and parallels the use of Unix.

Data base manager — More and more, an existing data base management system is being ported. Stratus is using Oracle Corp.'s Oracle, and Tolerant is using IBM's SQL. I have heard of others, as yet unconfirmed.

All of this flurry of activity is not without its problems. Several of the new start-ups have been delayed in getting their product to the marketplace. And DEC still has not delivered its promised VAX cluster software. But the march goes on, and the breadth of available fault-tolerant computing systems grows with the months.

When to consider

When does it make sense for you, the user, to consider going to a fault-tolerant system?

The answer is simple. Compare the cost of failures of your current system to the cost increment of going to a fault-tolerant system. The cost of a failure is the cost of lost business, idle workers and perhaps contingent liabilities incurred while the system is down. The downtime of a system is that time it takes to repair the hardware plus the time it takes to repair a corrupted data base, which could be many times the hardware repair time.

For new systems in which conversion is not a problem, the incremental cost of fault tolerance is rapidly approaching zero. After all, if a multiprocessor system is sharing a load across all its processors and if the overhead imposed by fault-tolerant considerations is minimal, a fault-tolerant system should be just as price/performance efficient as any other system. A look at today's offerings will verify that. And you get invaluable expandability as a freebie.

About the only reason not to go to fault tolerance for new systems today is because of management's concern for vendor stability. Some boards of directors don't want to hear anything but Big Blue, and Big Blue doesn't have a fault-tolerant system. But even this argument is dissipating as granddaddy Tandem rapidly approaches becoming a billion-dollar company.

Why haven't the major manufacturers entered this market? Good question with hazy answers. First, the perceived size of the marketplace may not yet be attractive. IBM certainly has proven that the personal computer market is a hotter trail to follow, at least for itself. Second, there is always the nagging problem of competition with existing product lines. And third, I am sure that the major manufacturers recognize the significance of the technological pitfalls in such a product development.

All that notwithstanding, I look at the development of fault tolerance in light of other breakthroughs that were major in their time, but are commonplace today. In the field of minicomputers, for example, 8K bytes was a lot of memory 15 years ago. Now, megabyte memory capabilities are taken for granted. Ten years ago, Fortran was about the only high-level language available for minicomputers, and multiuser operating systems were still unavailable. Would you even consider buying a mini today if it did not support many languages in a multiuser interactive environment? Of course not.

And so it goes with fault tolerance. As fault tolerance becomes more available as a near-zero cost option from major manufacturers, with unlimited expansion as an added plus, more and more new systems will be built as fault tolerant. As with memory and languages and operating systems, fault tolerance will become a subconscious requirement. We just wouldn't think about building a system without it.

How long? Anybody's guess. But based on past experience, it's got to be less than 10 years.

About the author

Wilbur H. Highleyman is chairman of The Sombers Group, Inc., Mountain Lakes, N.J., which specializes in turnkey software packages for on-line systems. Highleyman has nearly 30 years of experience in the development of real-time, on-line data processing systems, with particular emphasis on multiprocessor fault-tolerant systems and large communications-oriented systems. Highleyman is also founder and chairman of Minidata Services, Inc. and president of the Northeast Region Tandem User's Group.

© 2006 Sombers Associates, Inc., and W. H. Highleyman
www.availabilitydigest.com