the Availability Digest

All About Continuous Processing Architectures

October 2006

Achieving 100% Uptime

It is becoming trite to say that today's global economy and the enterprise's absolute reliance on computing technology mandates that critical IT services are never impaired – ever. However trite it may be, this statement is true. As Anton Lessing, CTO of BankServ said, "I am not interested in 9s. I am interested only in 100% uptime."

How can we achieve 100% uptime? Clearly, we can never know that we have achieved it – something that we did not anticipate can always happen decades down the line. However, we can come arbitrarily close to this goal.

Of course, "arbitrarily close" translates to "arbitrarily expensive." There is a limit to what we are willing to spend for extreme availability. Our willingness to spend is often related to the cost of downtime. Therefore, there is an omnipresent compromise between availability and cost. Actually, to be more accurate, one must compromise between availability, cost, and performance. Increasing availability will almost certainly have a negative impact on either cost or performance, perhaps even on both.

The class of architectures that can provide these extremely high levels of availability are known as *continuous processing architectures (CPA)*[1]. There are several ways in which such availability can be achieved through various CPA architectures, including

- active/active application networks,
- lock-stepped processors, and
- synchronized process pairs.

These architectures all have some characteristics in common:

- They are all redundant. Most uses today employ a single spare (dual modular redundancy, or DMR), though some provide two levels of sparing (triple modular redundancy, or TMR).
- They all provide fast failover, typically in the subsecond to second range.

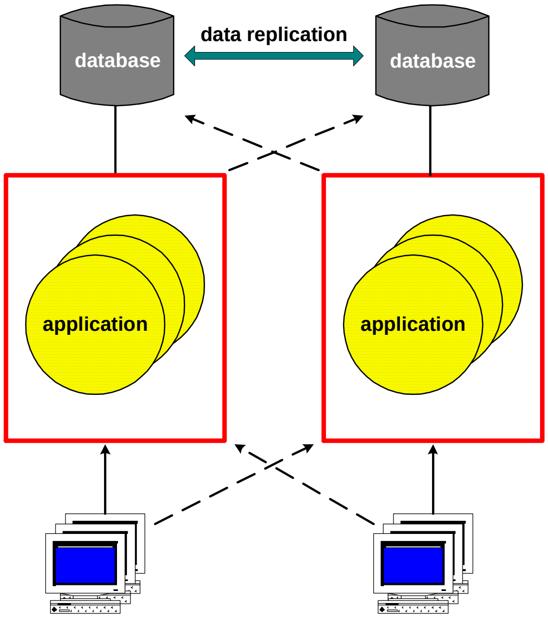
CPA does not eliminate component failures. In fact, we must accept that processors, disks, and networks will fail. What CPA does is to provide *fast failover*. If the downtime of a system following a failure is so short that it is not noticed by the users, in effect, there has been no downtime.

Each of these architectures is described below, along with examples of their current uses.

[1] The term "Continuous Processing Architectures" was coined by Jim Johnson of The Standish Group.

Active/Active Application Networks

An active/active application network comprises two or more independent nodes, each running the same application. Each node has access to at least two copies of the application database over a (hopefully) redundant network. Users can also communicate with any node over this network. To the extent that the nodes are geographically distributed, the application network is disaster tolerant, as an event that would take out one node would leave the others operational.[2]



Active/Active Application Network

Should a node fail for any reason, the users of that node can be switched to surviving nodes, typically in a subsecond or second time frame. Thus, the users on the failed node may not even be aware of the fault; and users on other nodes are totally unaffected, except for perhaps a slightly degraded response time due to the increased loading on their respective systems (depending upon the capacity of the nodes).

The database copies are kept in synchronism by replicating changes made to any one of the databases to the other database copies in the application network. There are several techniques for replicating this data:

- network transactions
- asynchronous replication
- synchronous replication

Active/active architectures are the new guys on the block and have only come into use with the commercial availability of high-speed data replication products. However, there are now several instances of active/active application networks, especially in large financial applications using NonStop systems (whose users, of course, are particularly interested in maximizing their uptimes).

Depending upon the type of system used for the nodes (i.e., high-availability or fault-tolerant), active/active systems can provide six to eight nines of availability (an average of 5 to 500 milliseconds per year of downtime), which translates into uptimes measured in centuries.

Active/active architectures are more fully described in our article entitled What is Active/Active?

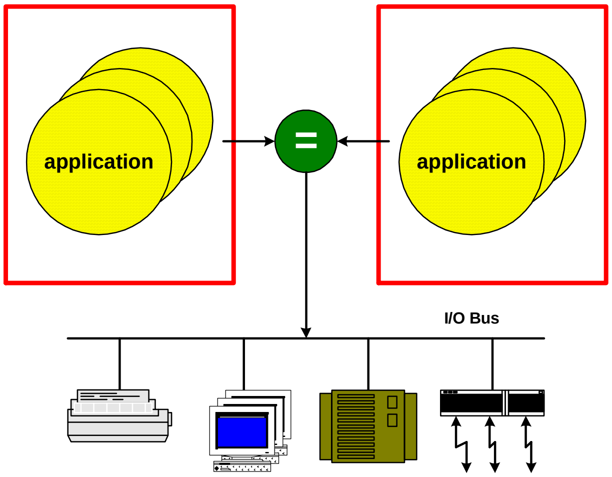
Lock-Stepped Processors

Lock-stepping involves keeping two processors in agreement by periodically comparing their outputs. Should there be a disagreement, the processor pair is immediately shut down. This is called *fast fail*. Fast fail has one significant advantage – it protects memory-resident structures or databases from data corruption caused by a sick processor.

Comparison can be done at any of many logical levels. For instance, memory writes could be compared. If one processor is trying to write into memory data that is different than what the other processor is trying to write, something is wrong; and the processor pair is shut down.

[2] See the book entitled *Breaking the Availability Barrier: Survivable Systems for Enterprise Computing*, by Dr. Bill Highleyman, Paul J. Holenstein, and Dr. Bruce Holenstein, published by AuthorHouse; 2004

Let us call a lock-stepped processor pair a logical processor. Of course, if a logical processor fails, there must be another logical processor to take its place in order for the system to continue in operation. This might be done, as exemplified below, by running multiple processors in a symmetric multiprocessing (SMP) environment or by keeping a companion processor synchronized and ready to take over via synchronized process pairs, as described in the next section.



Lock-Stepped Processor Pair

Lock-stepping is probably one of the oldest continuous processing architectures. For instance, it was used by the New York Racing Association at their Aqueduct, Belmont and Saratoga racetracks to implement in the early 1960s what was the first computerized totalizator system. A totalizator is the system used by race tracks to sell tickets, calculate odds and payoffs, and display the results on the large infield board and on other displays around the race track. This system used a pair of Honeywell H200 computers.

Old as it may be, lock-stepping is in current use today in commercially-available, fault-tolerant systems. Stratus uses lock-stepped processor pairs in their ftServer series of fault-tolerant systems. These processor pairs check each other at the memory-write level. ftServer systems run as a set of SMP logical processors. Should a logical processor fail, there are others that will pick up the load.

HP uses lock-stepped processors in its fault-tolerant NonStop servers. In HP's earlier K-series and S-series systems, logical processors comprised a pair of processors coordinating at the memory-access level. In that case, each critical process was backed up by maintaining synchronization with a companion process in another logical processor. Should a logical processor fail, the backup process took over immediately, with no impact on other processes using it.

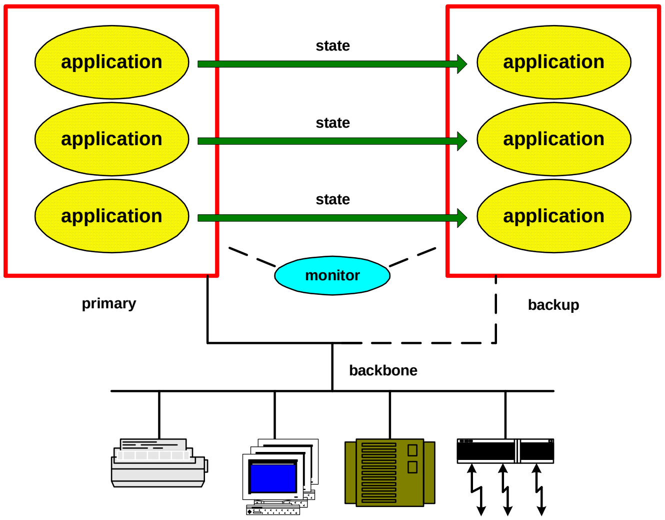
In its new Integrity line of fault-tolerant systems, HP again uses lock-stepped processors as logical processors. However, in this case, coordination between the two physical processors is done at the message level. Whenever a processor is ready to send a message to the outside world (done over ServerNet), it waits until the other processor is also ready to do so. The two messages are compared. If they are the same, the message is released. Otherwise, the logical processor is failed; and the backup processes take over.

There is an added benefit to this looser level of lock-stepping. If one processor fails, and if this failure is detected by the processor's extensive internal error-checking logic, the processor will take itself out of service. The other processor is then free to continue providing service. The only thing that is lost is the guarantee of data integrity – the highly unlikely event that this processor could make an error and contaminate memory data structures or the database.

HP not only offers DMR, as described above, but also a TMR configuration in its Integrity series. In this case, each logical processor can contain three physical processors. Should a message mismatch occur, the bad processor is voted out; and the logical processor continues on as a DMR pair with full data integrity protection. With this configuration, HP has essentially removed processor failures as a factor in the availability equation.

Synchronized Process Pairs

A synchronized process pair comprises a primary process (a program running in a processor) and a backup process. The primary process is providing all processing functions, and the backup process is idle.



Synchronized Process Pairs

Whenever the internal state of the primary process changes, the new state is sent over a high-speed channel to the backup process, which updates its state. Should the primary process fail, the operating system automatically routes all new requests to the backup process, which continues processing with no interruption to other processes using this process pair.

Synchronized process pairs have been used for all critical processes, such as the disk and communication processes, in NonStop systems since the early days (when they were Tandem systems). They continue in use today to recover from logical processor failures.

Combined Technologies

Though many of these technologies are very old – lock-stepped processors date back over forty years and synchronized process pairs over thirty years, they are still in active use today. In fact, as described above, NonStop systems still use both technologies.

However, neither of these architectures is supported by off-the-shelf, commercially available products. Consequently, they generally are not suitable for implementing user applications. They are relegated to the important function of implementing fault-tolerant systems.

Enter active/active. To build an active/active system, all one needs is a high-speed data replication engine. These are now available as off-the-shelf products from several vendors. Though IBM provides data replication as part of its Parallel Sysplex systems, most other vendors do not. They depend upon the products of third parties.

Especially in the NonStop world, where such products are plentiful, active/active technology is taking hold. Interestingly, in a NonStop active/active application network, all three of these technologies are used. Lock-stepping is used in the logical processors, synchronized process pairs are used to back up logical processors, and active/active nodes are used to obtain extreme availabilities.

100% Uptime Achieved - Almost

As noted above, active/active systems can provide uptimes in the order of centuries; and that is based on there being just one spare node. If there are two spare nodes, availabilities of nine to twelve 9s can be achieved. This is as close to 100% uptime as we need to get with active/active technology since other unanticipated factors are now more likely, such as the 2004 Northeast blackout in North America.