the Availability Digest

CPA at Aqueduct, Belmont, and Saratoga Race Tracks

January 2007

Continuous processing architectures are nothing new. They have been around for decades and used the same techniques that we use today. We take a look in this article at one of the early continuous processing architectures, implemented and put into production over four decades ago.

The New York Racing Association

The New York Racing Association (NYRA) operates some of the best known race tracks in the world – Aqueduct, Belmont, and Saratoga. Belmont is the home of the Belmont Stakes, one of the three racing events that together are known as the Triple Crown.

Totalizator Systems

All race tracks, including NYRA, operate *totalizator* systems. A totalizator is the system which accumulates wagers into pools. Based on the monies accumulated in each pool, it calculates and displays the current odds that each horse in the race may come in first (win), second (place), or show (third) and displays the payoffs for the winning horses at the conclusion of the race.

Up until the early 1960s, totalizators were electromechanical relay systems. These systems were inflexible in that adding additional wagering types such as a Daily Double (choosing the first and second place horses) or a Trifecta (choosing the top three winners) was very difficult. In addition, they were not as reliable as would be desired and required continual maintenance by highly skilled technicians.

In the early 1960s, NYRA decided to build a computerized totalizator system to solve these problems. This had never been done before, so NYRA was the pioneer.[1]

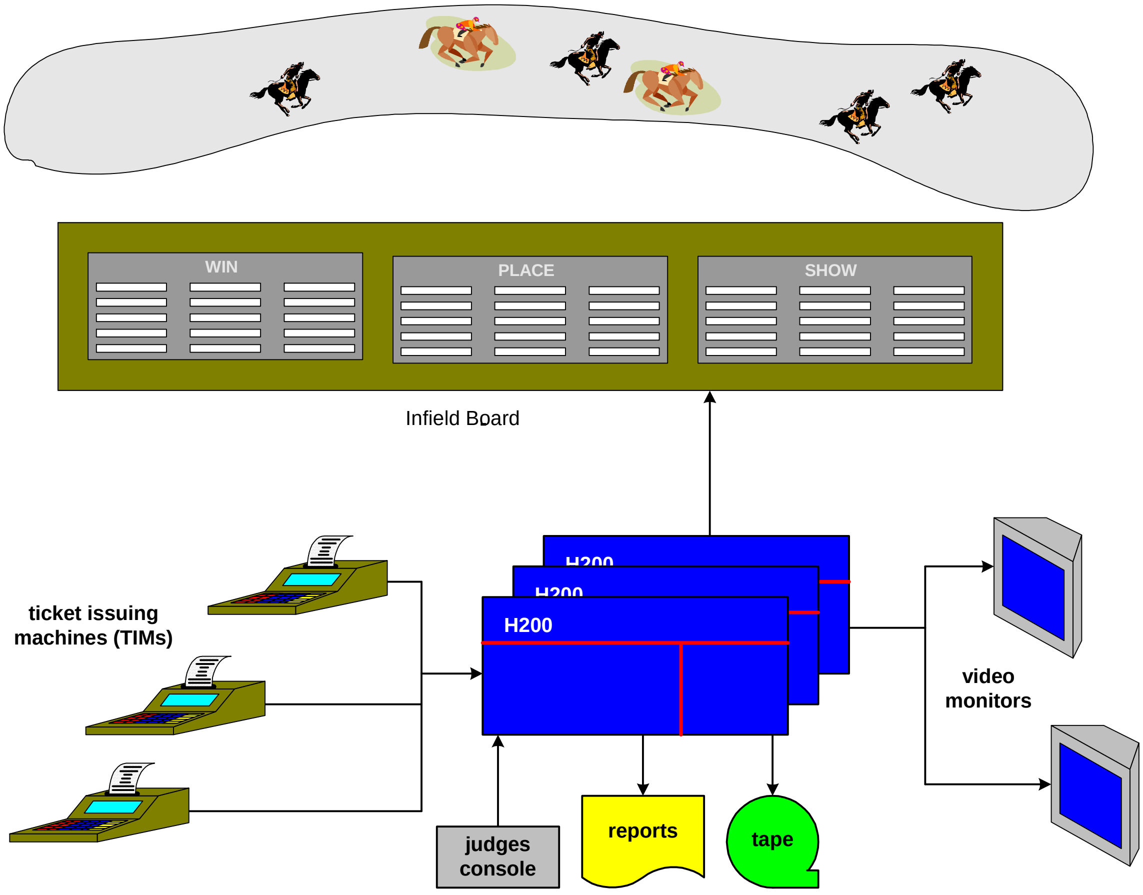
System Functions

This first computerized totalizator system was to perform all of the functions of its electromechanical kin. It needed to service 600 ticket issuing machines (TIMs) generating 1,200 transactions per second (a hefty transaction rate even by today's standards). Win, Place, Show, and Daily Double pools were to be supported; and the system had to be expandable to handle other types of wagering (a Trifecta was later added). In addition, advanced sales on later races had to be supported.

[1] This was proven in a lengthy patent case brought by Westbury Race Track in Queens, New York. Westbury claimed also to have been first.

A ticket transaction was fairly simple. It included the race number, the horse number (or numbers for the Daily Double), the pool (Win, Place, Show, Daily Double) and the amount wagered. Against all good design at that time and today, each transaction was processed sequentially in the interrupt routine. In this way, ticket processing time could be kept to less than 800 microseconds, the maximum processing time allowed to achieve the 1,200 transactions per second throughput rate.

Each wager was added to the pool of money accumulating for each horse in the specified pool. Every ninety seconds, the odds of a horse coming into the money in each pool were calculated based on the horse's popularity in the betting. These odds were posted to the large infield board and to monitors distributed around the race track. In addition, the current state of each pool was printed and written to magnetic tape for later auditing purposes.



NYRA Totalizator System

Also calculated and displayed were the anticipated payoffs based on these odds after deducting the state's take from the pool and rounding down the payoff to the nearest five cents on the dollar (the remainder was called breakage and also went to the state). If a horse was predicted to pay less than five cents on the dollar, the payoff was rounded up to that amount – a difference coughed up by the track (great care was taken to choose horses for a race in which there was no obvious favorite so as to avoid this situation).

Wagering stopped when the horses left the gates. At the conclusion of the race, the judges posted the winners by entering the finishing order of the horses via a Judges Console. The system then calculated and displayed the final payoffs. It also printed and wrote to tape a variety of reports showing the final pool amounts, the number of tickets sold for each horse in each pool, the payoff amounts, and the state's take for its percentage of pools and the breakage.

Availability Requirements

The availability of the totalizator system was a critical issue. Consider the following scenario - a bettor is convinced that horse #3 is going to win against all odds and will provide a big payoff. As the bettor stands in line shortly before the race with $100 to wager, the totalizator system fails. No more tickets can be sold.

The race is run anyway based on pools which existed before the system failure, and horse #3 does win. The payout is 50:1. Had the totalizator system been in working order, the bettor would have won $5,000. Instead, he walks away with only the $100 he was unable to wager. Similarly, hundreds of other race track patrons are also cheated out of potential winnings.

This situation had happened in the past, with resulting riots and significant damage to race track facilities. Therefore, totalizator failure had to be avoided at all costs. 100% uptime was paramount. The Recovery Time Objective (RTO) was zero.

In addition, the system had to be protected from data loss should it fail. The pools still had to be accounted for, and monies still had to be paid out. The Recovery Point Objective, the measure of the amount of data that could be lost due to a system failure, was also zero.

System Architecture

The system chosen for this task was the Honeywell H200 (yes, Honeywell was a major computer supplier back then). Interestingly, the H200 was a six-bit character-oriented machine, making it an unlikely system for such requirements. However, it was blazingly fast and had an advanced interrupt system for its time. It was deemed to be one of the few systems capable of handling the transaction workload.

The prime contractor was Australia-based Automatic Totalizator, Ltd., which won out over the American Totalizator Company. The information for this case study came from the files of Dr. Bill Highleyman who was system architect and project manager for the system. Dr. Highleyman is now Managing Editor of the Availability Digest.

Triple Modular Redundancy

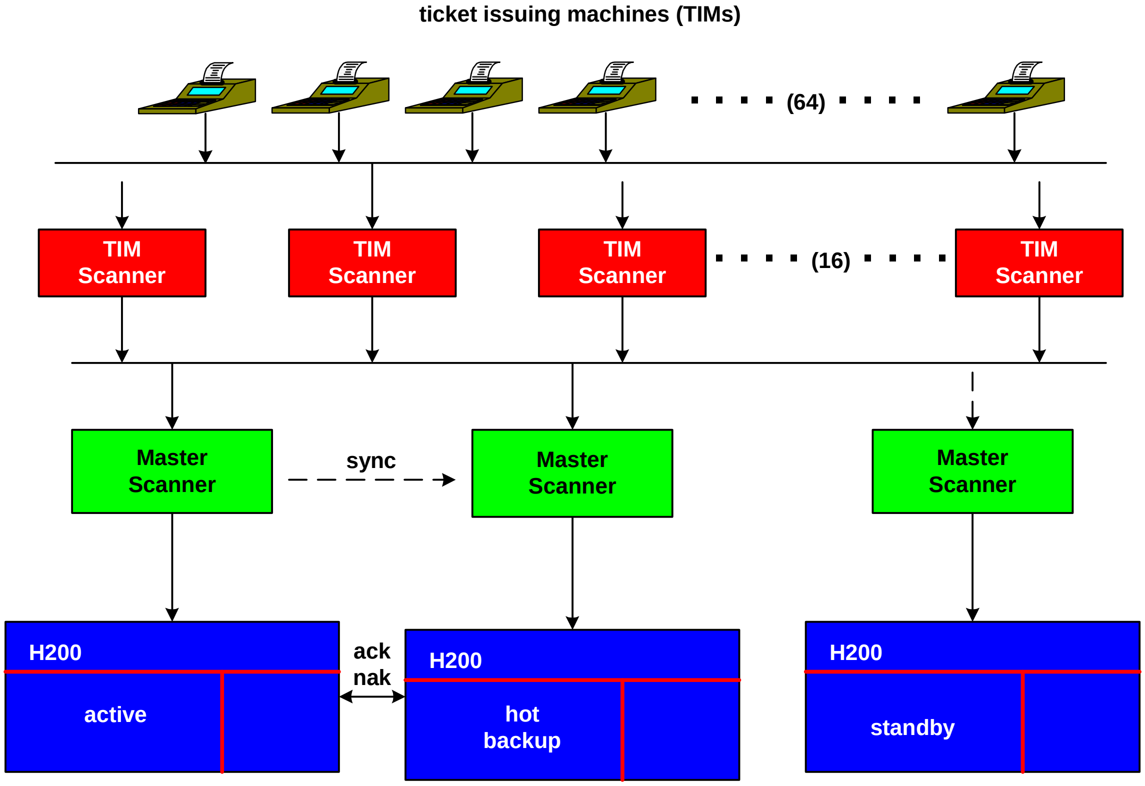
The system was designed to be a triplexed system (triple modular redundancy). Three identically configured H200 systems cooperated in the operation. One acted as the active system, one as a hot backup, and one as a cold standby.

The active system and the backup system were maintained in synchronism via lock-stepping at the transaction level. Should the active system fail, an operator command switched roles; and the backup system immediately took over the active role. The only delay in system recovery was the operator's decision time to switch over.

Should either the active system or the backup system fail, the cold standby was put into service as the new hot backup system. It was synchronized with the current active system by transferring the current state of the pools via a memory-to-memory transfer, a process which took only a few seconds and was done concurrently with continued wagering.

Transaction Processing

The active and backup systems were kept in synchronism by having each system process each transaction – primarily ticket transactions and the Judges Console inputs. This was accomplished by using a lock-stepped TIM scanning complex.



Totalizator Transaction Synchronization

The scanner complex comprised two levels, Master Scanners and TIM Scanners. Each of the three H200 systems had its own Master Scanner. The Master Scanners could poll up to 16 TIM Scanners. The TIM Scanners polled the ticket issuing machines. Each could poll up to 64 TIMs, giving a system capacity of 1,024 TIMs, more than the required 600.

At a ticket issuing machine, the selling clerk entered the ticket details (race number, horse, pool, and amount) and pushed the sell key. This caused the TIM to generate a request signal to its TIM Scanner.

Each TIM Scanner scanned its TIMs for request signals. When it found one, it stopped and locked onto that TIM, downloading the TIM's transaction data into its own buffers. It then generated a request to the Master Scanners.

The Master Scanner for the active system scanned all TIM Scanners for an active request. When it found one, it locked onto that TIM scanner, loaded the transaction data into its own buffers, and generated an interrupt request to the active system. At the interrupt level, the active system read the transaction data from the Master Scanner, processed it, and generated either an Acknowledge indication (ack) or a Reject indication (nak).

The Acknowledge/Reject indication was returned through the scanner complex to the TIM, which either printed the ticket or illuminated a rejection indicator for the selling clerk. At that point, the involved TIM Scanner and the Master Scanner were released to search for other requests.

The only other interrupt activity of any note was that required to process print and tape buffers every ninety seconds. This had little impact on ticket selling capacity.

Database Synchronization

It was, of course, important that the active and backup systems had the same pool data in their databases. This was accomplished by having the systems independently process each transaction and then compare their results. The results to be compared were simply the acknowledge or reject decisions.

To ensure that the two systems were in lock step so far as which transaction was being processed, the active Master Scanner controlled the transaction processing sequence. When it found a request from a TIM Scanner, it notified the backup Master Scanner as to which TIM Scanner to service next. The backup Master Scanner then obtained the same transaction information from the selected TIM Scanner and processed it.

To synchronize the results, the active and backup systems compared the resulting Acknowledge/Reject indications. If they agreed, the active Master Scanner released that indication to the TIM Scanner. If they disagreed (i.e., one system wanted to acknowledge the transaction, and the other wanted to reject the transaction), they both rejected the transaction; and a reject indication was sent to the TIM via its TIM scanner.

Thus, in effect, the totalizator system was a triply redundant system, which maintained synchronization between its active and backup systems by lock-stepping at the transaction level.

Fault Detection

The failure of a critical component was detected within a fraction of a second. This was done via two means.

First, the TIM Scanners and active Master Scanner sounded an alarm and illuminated a fault indicator if they did not have a request serviced within a few milliseconds. Since transaction processing was in the order of a millisecond, and since there were 16 TIM Scanners serviced sequentially, a TIM Scanner request should have been serviced within 16 milliseconds. A Master Scanner request should have been processed immediately since it was serviced at the interrupt level by the active system. To provide a level of tolerance, these fault timers were set to a value in the order of 100 milliseconds.

The second fault detection means was implemented in the active and backup systems. If one system wanted to acknowledge a transaction, but the other system wanted to reject that transaction, the transaction was rejected. However, the system desiring to acknowledge the transaction incremented a fault counter for the other system. If a fault counter reached a specified value, the system rejecting the transactions was reported to be down. Similarly, if one of the systems did not respond with a transaction ack/nak, it was reported to be down by the other system.

Failure recovery was straightforward and required little thought. If a scanner alarm sounded, or if the backup system reported that the active system had failed, the operator pressed the button that activated the backup. It was assumed that the active system with its Master Scanner had become incapable of servicing the scanning complex.

If a TIM Scanner alarm continued to sound, then the fault was in that TIM Scanner; and it was taken out of service. In any event, if it was determined that the active system had failed, the cold standby system was brought into service as the new backup system.

Failover Testing

A key issue in achieving high availability was the switchover and recovery time. In this case, the switchover time was very fast because a hot backup system was used. The main component of switchover time was the time required for the operators to recognize that there was a problem. At that point, it took only the pressing of a button to effect switchover, which was totally automated.

Recovery of the system to a duplexed state from a failure was much more complicated. Applications had to be loaded into the standby system (which typically was being used for other administrative functions). The standby system then had to be activated as the new hot backup system, and the current pool database had to be transferred to it via a memory-to-memory transfer. This database transfer occurred during active wagering. If a transaction arrived for an element in the pool database that had not yet been transferred, the transaction was processed anyway by the backup system. The value for the affected data item was corrected when that data element was copied from the active system.

NYRA and the operations staff were very serious about keeping switchover and recovery smooth and rapid. To do this, there were weekly exercises prior to the opening of the race track to the public. A team of TIM clerks simulated betting, and the chief operator devised various failure scenarios to train the operations staff. One result of this level of training was that every operator was trained to quickly recognize any system fault and to take corrective action. Upper management never had to be consulted to execute a switchover or recovery action. The recovery strategy was “If in doubt, straighten it out.”

Cutover

Live cutover was very critical as the race track dealt with real money from real customers. The system was first put into operation at NYRA’s Aqueduct race track. As a hedge, NYRA offered several days of racing at no cost to the public. Wagering was on paper only – no money actually changed hands. The old electomechanical totalizator was kept poised to take over in the event of a failure of the new system, which, in fact, had to be done on at least one occasion.

Only when the new system had proven itself reliable after several days of service was the old totalizator decommissioned.

Security Enforced by a Distraught Wife

Security was another serious issue. Every attempt was made to ensure that there was no cashing of fraudulent tickets. The tickets were printed on a paper that could not be modified. Each had a code on it identifying the race, and the code was not divulged beforehand.

The security protections worked very well until a later version of the system using Digital PDP-8s was introduced. These systems had a front panel with switches that could be used to pause the computer, modify data, and restart the computer. This should not have caused a problem since there were several people in the computer room, and any such activity would have been noticed.

Unless everyone was in on the fraud. And that is exactly what happened at a small race track. The computer room was staffed by a small group of two operators and a TIM technician. The gambit was to wait for the race to be completed and for the judges to declare the winners. One operator then stopped the backup system, added one or two tickets for a high-paying winning combination such as the Trifecta, and then forced a switchover. The modified system calculated the ticket payoffs, which now included the fraudulent tickets. In effect, some money was taken out of the pockets of the real winners, but this amount was a very small fraction of the total pool.

Meanwhile, the TIM technician, having installed the race code in a test TIM in the computer room, printed offline the number of fraudulent tickets that had been entered into the computer. These tickets were then given to runners who cashed the tickets for a fee, not knowing that they were fraudulent.

All of this fraudulent activity was reflected in the final printed reports so that the auditors never caught on. The scheme was successful until the wife of one of the operators caught him with the wife of one of the other perpetrators. The distraught wife blew the whistle, and three young men spent some time in prison.

Postscript

This early, continuously available system remained in operation for almost two decades. It morphed into several later versions of very successful totalizator systems installed at hundreds of tracks around the world. Following the H200 system was one based on Digital PDP-8s, then Tandem systems, followed by Digital PDP-11s.

The original electromechanical TIMs were replaced with intelligent microprocessor-based TIMs. Wagers were made via mark-sense cards, which were filled out by the bettors and which supported a wide range of new, exotic wagering products. The TIMs produced machine-readable tickets, each of which contained a unique, bar-coded serial number that was recorded in the database. Each ticket could contain a multiplicity of wagers.

Ticket cashing was done automatically by reading the ticket and then looking up the payoff amounts. The ticket serial number was checked to ensure that it was a valid ticket. Security provisions were instituted in the computer room to prevent any future fraudulent collaboration.

In later years, both the Tandem systems and the PDP-11 (now VAX) systems morphed into state lottery systems.