the Availability Digest

www.availabilitydigest.com

Apollo 11 – Continuous Availability, 1960s Style

September 2009

We talk a lot about continuous availability – systems that virtually never fail – as if this is a brand new concept made possible by today's technologies of fault-tolerant systems, clusters, virtualization, and active/active systems. We talk about task-critical systems, in which a fault will idle workers or production lines, mission-critical systems that can take down a company, and safety-critical systems whose failure can mean loss of life or property.

But forty years ago, continuously available systems put men on the moon and got them back several times without a loss. The entire lunar mission, from launch to translunar coast to landing and then rendezvous and return to earth, were controlled by computers whose failure would mean the loss of the astronauts. Against all odds, all systems worked impeccably for the lunar landings.

How did the technologies of the '60s achieve this near-miracle? In this article, we look at the approaches taken back then to ensure continuous availability. We will find that they were not much different than what we do today. The ability to achieve continuous availability in our computer systems has been around for a long time.

The Apollo Mission

On April 12, 1961, the U.S. was galvanized into action when the Russians put Yuri Gagarin into orbit in the Vostok 1. The U.S. was clearly behind in the space race.[1]

Shortly thereafter, President John F. Kennedy threw down the gauntlet:

> "I believe that this nation should commit itself to achieving the goal, before this decade is out, of landing a man on the Moon and returning him safely to the Earth. No single space project in this period will be more impressive to mankind or more important in the long-range exploration of space, and none will be so difficult or expensive to accomplish."

This seemingly impossible goal was met on July 20, 1969, when Neil Armstrong stepped off of the lunar lander's ladder onto the moon and said "That's one small step for [a] man, one giant leap for mankind."[2]

This "giant leap" required an investment of $24 billion (in 1960 dollars), 400,000 people, and 20,000 industrial firms and universities over an eight-year period. The costs of the spacecraft and

---

[1] Project Mercury was authorized under President Dwight D. Eisenhower in 1958 to put a man into space; but this goal was not achieved until John Glenn orbited the earth three times on February 20, 1962.

[2] It is reported that Armstrong meant to say "a man" but left out the article "a."

rockets added another $83 billion to the tab. The mission continued to put men on the moon five more times, ending finally in 1972.[3]

The Apollo Computers

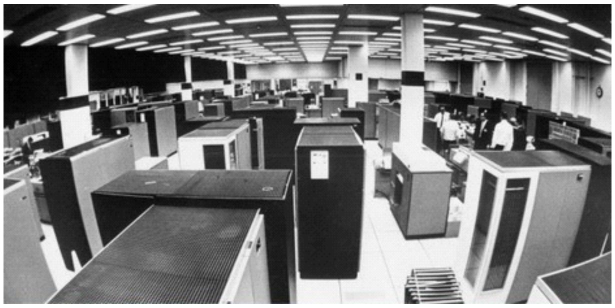
Three main computer complexes were involved in each lunar flight – the ground computers, the onboard guidance computers, and the rocket computer.

Ground Computers

The *Availability Digest* was fortunate to have the opportunity to speak with Ed Poole, an engineer who contributed to the Apollo ground-computer complex during the mission. He joined NASA in 1966 with a mathematics degree, a necessity for the complex navigation calculations required to guide men to the moon. The following is a description of the ground-control environment that Mr. Poole provided us.

As he related, the overriding design principle was to be “man-ready.” The entire thrust of the mission was to successfully put a man on the moon without killing anybody.[4] After all, the astronauts were friends and neighbors. “Man-ready” revolved around redundancy and extensive testing.

The effort wasn’t easy, nor was it cheap. Redundancy was achieved through the use of five newly-released IBM 360 Model 75 computers running the MVS operating system. Up to 3,500 IBM employees participated in the design, implementation, and testing of the system.



The Apollo 11 IBM Computer Room

To put technology into perspective, at that time the computer industry was just in the process of converting from vacuum tubes to transistors. IBM had delivered its first transistorized computer, the IBM 7090, in 1959. The computers received by NASA in 1964 were among the first IBM 360s delivered. The new MVS operating system was a batch-oriented system but had to do a real-time job.

The 360s had a speed of 1.7 MIPS (million instructions per second) and contained one megabyte of memory. Memory was later expanded to two and then four megabytes per computer.

The programming staff was divided into three groups whose jobs were to:

- prepare software specifications and test plans.
- modify MVS to provide real-time processing.
- specify error-recovery procedures.

Because the 360 and MVS were batch-oriented, one of the first tasks of the programming group was to modify them for real time. This required gaining access to the 360 bus structure, a task that proved to be difficult. Specialized communication plug boards had to be connected to the I/O bus to receive ground tracking, onboard telemetry, and trajectory data in real time. The plug

---

[3] Apollo Program, *Wikipedia.*

[4] Unfortunately, astronauts Gus Grissom, Ed White, and Roger Chaffee were killed in 1967 when their Apollo 1 command module caught fire on the launch pad during a training exercise.

boards interfaced to the computer bus via a five-way switch so that they could be connected to any of the five computers.

As Mr. Poole told us, these computers had to do error recovery in ways no one had ever considered before. It was for this reason that a special team was devoted to error detection and recovery. Mr. Poole worked with this team to provide simulation of any conceivable fault to validate the error-recovery procedures. Communication errors were handled by a 3-bit error correcting code using a 36-bit polynomial. Uncorrectable messages were retransmitted. Trajectory errors had to be corrected in real-time. If there were to be an error in the onboard guidance systems, this had to be accommodated from the ground since the guidance computer programs could not be patched – they were held in ROM.

Simulations were also implemented to train astronauts and controllers.

At the time, IBM described the six megabyte programs it developed to monitor the spacecrafts' environmental data and the astronauts' biomedical data as the most complex software ever written.[5]

Redundancy was provided via the use of three of the five IBM computers working in parallel. The other two computers were available for test and development but could be pressed into production if one of the three production computers went down.

Of the three production computers, one was the operations computer, one was a dynamic standby, and one was a static standby. The operations computer and the dynamic standby processed all data in parallel. If one went down, the other immediately and automatically took over control and became the operations computer.

At this point, the static standby computer was brought into service as the dynamic standby. It was brought up-to-date by loading the last data checkpoint and then by replaying all subsequent events at high speed. Checkpoints were written to tape every fifteen minutes to be used to load the static standby, and real-time events were written to tape as they occurred. The event tape was used to complete the resynchronization of the new standby.[6]

During this recovery time, operations continued unimpeded. A total system failure would have required that following a failure of one of the active computers, the other would fail before the new standby was synchronized. Even then, only a small amount of data would have been lost; and the system would quickly resynchronize with newly received data. Continuous availability was truly achieved.

There was a lot of demand for computer time. Therefore the static standby was in place only during the critical phases of launch, translunar coast (during which the space capsule was thrown out of earth orbit and into lunar orbit), lunar landing, and return to earth. During other times, the static standby system was available for other purposes.

Guidance Computers

The onboard Apollo Guidance Computer (AGC) controlled the Columbia Command Module (CM) and the LM Guidance Computer (LGC) controlled the Eagle Lunar Module (LM). They collected flight information, displayed this to the astronauts, and communicated it to the ground. The AGC was developed by the MIT Instrumentation Laboratory and was built by Raytheon. There was one AGC in the Command Module and one in the Landing Module.[7] Those with a need to know

---

[5] Apollo 11: The computers that put man on the moon, *Computer Weekly*; July 17, 2009.

[6] Interestingly, this same architecture was used at about the same time to run the totalizator system for the New York Racing Association. See CPA at Aqueduct, Belmont, and Saratoga Racetracks, *Availability Digest*; January 2007.

[7] Apollo Guidance Computer, *Wikipedia.*

carried a green (AGC) or a yellow LGC handbook in order to issue manual command interface updates to these computers. The guidance systems provided for manual control should an AGC exhibit problems.

The AGC ran the LM's Primary Guidance, Navigation, and Control System (PGNCS). The PGNCS was backed up by the Abort Guidance System (AGS) designed by TRW. The AGS could be used to take off from the moon and rendezvous with the CM, but it could not be used for landing.

The Command Module

The AGC was the first computer to use integrated circuits (ICs). All 4,100 ICs were identical three-input nor (negative or) gates. They were configured to provide flip-flops for data storage and to provide the logic of the computer. The ICs were socket-mounted and were interconnected by wire wrap. The decision to use a single IC design throughout the AGC avoided problems that plagued other early IC designs that used a mix of IC technologies.

The Apollo project perhaps kick-started the microchip revolution. It bought 60% of the American commercial production of ICs in the early 1960s, temporarily shoring up an industry for which few other markets yet existed.[8]

The memory of each of the AGC lock-stepped computers comprised 2K words of random-access RAM memory and 36K of read-only ROM memory, each with a cycle time of about 12 microseconds. The word length was 16 bits. The processor ran at a speed of one megahertz and could multitask eight jobs at a time via time slots.

Modern-day PCs may be more powerful than the AGC, but the AGC did much more than plug into a printer and a router. It interfaced with ground telemetry links, radar rendezvous systems, landing altimeters, gyro compasses, optical star trackers, and propulsion systems.

The Landing Module

Unfortunately, the PGNCS malfunctioned on the first lunar landing. It became overloaded with radar data from spurious signals generated by the rendezvous radar that had been left on during descent. Fortunately, the efforts at automatic error recovery paid off. The PGNCS shed some noncritical tasks and returned to service in time for a safe lunar landing.

The Rocket Computer

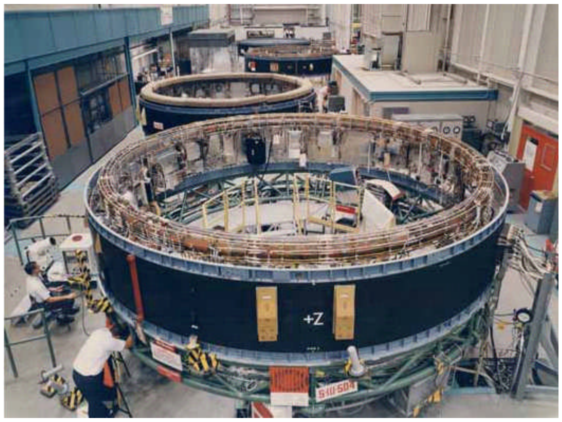
The final computer in this trilogy is the one that guided the Saturn 5 rocket at liftoff. Designed by NASA and built and programmed by IBM, the Launch Vehicle Digital Computer (LVDC) was embedded within a three-foot high, 21-foot diameter ring that mounted onto the Saturn V rocket. It was the nerve center for the launch vehicle.[9]

The LVDC had a two megahertz clock rate and contained 32K, 28-bit words. For reliability, it used triple-redundant logic with voting.[10] Each logic system was split into a seven-stage pipeline. At each stage, a vote was taken; and the most popular result was passed on to the next stage in each pipeline. Even with this, calculations showed only a 99.6% reliability over 250 hours of operation. On the other hand, imagine the environment in which this computer had to operate.

---

[8] Apollo Guidance System revisited, *IET*; July 6, 2009.
[9] IBM Mainframe Assists Apollo 11 Landing, *IBM Systems Magazine*; July 2009.
[10] Saturn Launch Vehicle Digital Computer, *Wikipedia.*

The Saturn V Launch Vehicle



The Launch Vehicle Digital Computer Ring

After less than six hours of productive activity radioing onboard measurements back to Earth and computing guidance control and engine control, the LVDC completed its life work. The third-stage Saturn V rocket engine nosed down into the ocean, and the computer ring was turned loose to orbit the sun.

Summary

Lessons from the Apollo experience have been carried through the years and have contributed significantly to the development of today's mainframe and embedded computers. In addition, these lessons have been applied to subsequent space missions.

The space shuttle's specialized AP101 computer is a later generation of the AGC. It also uses five computer copies, each with sixteen I/O processors, all running in lock step.

The original space shuttle ground-control computers were a one-off of the Apollo architecture. NASA has since moved to high-performance clusters using IBM RS6000s.

The lunar program led to the development of safety-critical systems and the practice of software engineering to program those systems. Much of this knowledge gleaned from the Apollo program forms the basis of modern-day computing.

Ed Poole retired and bought hundreds of acres in Texas, where he runs a campground. He has since been called back to NASA and is contributing half of his time in Houston to tomorrow's space missions.