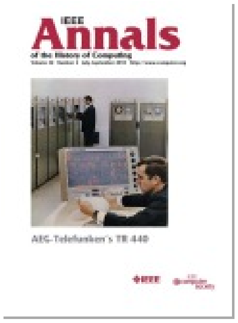
RAID: A Personal Recollection of How Storage Became a System

Randy H. Katz

IEEE Annals of the History of Computing, Volume 32, Number 4, October-December 2010, pp. 82-86 (Article)

Published by IEEE Computer Society



RAID: A Personal Recollection of How Storage Became a System

Randy H. Katz
University of California, Berkeley

Editor: Dave Walden

My colleagues David Patterson and Garth Gibson (then a graduate student) and I first defined the acronym RAID, or redundant arrays of inexpensive disks, in a paper we wrote in 1987.[1] The RAID idea was that it was feasible to achieve significantly higher levels of storage reliability from possibly very large numbers of lower-cost and lower-reliability smaller disk drives, which were then emerging for personal computers. It accomplished this by presenting the image of single logical disk, yet arranging the underlying physical disks into arrays. Data would be partitioned and redundantly spread across these. Figure 1 summarizes the case we made at the time.

We were not the first to think of the idea of replacing what Patterson described as a slow large expensive disk (SLED) with an array of inexpensive disks. For example, the concept of disk mirroring, pioneered by Tandem, was well known, and some storage products had already been constructed around arrays of small disks. At the time, however, it was not generally understood and appreciated was that there was a continuum of performance and capacity trade-offs in making large numbers of small disks a reliable alternative for organizing the storage system.

RAID was simply the right idea at the right time. Our taxonomy introduced a common terminology and conceptual framework that helped to galvanize the storage industry. Many firms immediately began developing RAID products. Of course, what started as a research concept was soon co-opted by that industry. They soon replaced the ''inexpensive'' in RAID, due to its low-cost implications, with the ''independent.'' Today, virtually all server- and networked-based storage is based on RAID, and even many PC users have hardware or software RAID systems set up on their machines.

What is RAID?

In 1987, a typical mainframe disk had a 14-inch diameter and a 7.5-GByte capacity. The best small disk drivers had a 3.5-inch diameter and a 100-MByte capacity. Just matching the capacity of the mainframe disk would require 75 small disks. Even though the small disk had lower performance than the large disk, in aggregate, the array of smaller disks could achieve a higher rate of I/Os per second and transfer rate. The problem was that the probability of a disk failing scales with the number of disks, so it is much higher for a nonredundant array than a SLED.

Our RAID taxonomy introduced a numbering scheme to distinguish between how redundancy is introduced and data is spread among the drives in the array. RAID 0 introduces no redundancy but spreads logical data across multiple disks by taking sequential blocks of storage and allocating them one by one across the underlying physical drives. RAID 1 is mirrored storage, creating one copy of each data block and placing it on a second drive, mirroring the primary copy. RAID 2 treats the disks within the array as if they were memory chips, interleaving data bitwise in parallel across the drives and using traditional Hamming codes to correct any failures should one of the disks fail. RAID 3 reduces the redundancy overhead to a single bit per bit position based on the realization that parity is sufficient to correct data loss if the failed disk's position is easy to determine, which it is since disks communicate with the rest of the system through high-level protocols. RAID 4 extends this idea by interleaving not at the bit level, but in units of larger blocks. Although all disks in the array must be read or written together in RAID 3, they can be accessed independently in RAID 4. Writes require special care to update the parity by complementing the bits that have changed between the old and new data blocks. Because all the parity redundancy is on one disk, it becomes a performance bottleneck on writes. RAID 5 extends this idea by interleaving the parity blocks as well as the data, as much as doubling the number of writes that could be supported. The RAID Wikipedia entry does a commendable job of defining the RAID levels in greater detail (see http://en.wikipedia.org/wiki/RAID).

In retrospect, our choice of RAID levels was a mistake. RAID 5 is not always better than the lower levels. For example, in a large block read and write environment, RAID 3 might be appropriate. This also created a minor industry in defining new RAID levels over the years.

Berkeley RAIDers

In 1983, I joined the Berkeley faculty from the University of Wisconsin, Madison. My colleague from Wisconsin, David DeWitt, spent the 1983/1984 academic year on sabbatical at Berkeley. We joined with Michael Stonebraker, leader of the Ingres Database Project, to study the ultimate limits to performance in database systems—that is, the upper bound on the number of transactions per second that could be processed. This led us to an investigation of main memory database

1058-6180/10/$26.00 © 2010 IEEE

	IBM 3380	*Fujitsu M2361A*	*Conners CP3100*	*3380 v. 3100*	*2361 v. 3100*
Characteristics				**(>1 means 3100 better)**	
Disk diameter (in)	14	10.5	3.5	4	3
Formatted Data Capacity (MB)	7500	600	100	.01	.2
Price/MB (cntl incl)	$18-$10	$20-$17	$10-$7	1-2.5	1.7-3
MTTF Rated (hrs)	30,000	20,000	30,000	1	1.5
MTTF in practice (hrs)	100,000	?	?	?	?
No. Actuators	4	1	1	2	1
Max IOs/s/Actuator	50	40	30	.6	.8
Typ IOs/s/Actuator	30	24	20	.7	.8
Max IOs/s/box	200	40	30	.2	.8
Typ IOs/s/box	120	24	20	.2	.8
Transfer Rate (MB/s)	3	2.5	1	3	4
Power/box (W)	6,600	640	10	660	64
Volume (cu ft)	24	3.4	.03	800	110

Figure 1. Table from the original 1987 RAID paper. We made the case for building storage systems from large numbers of small form-factor disk drives.[1]

systems. We determined the critical performance bottleneck was writing the transaction commit log.[2] Because the log had to be written to stable storage, which at the time meant disk, it planted the seed that we needed a breakthrough for achieving much greater I/O rates to punch through this bottleneck.

The development of the personal computer in the early 1980s pushed the development of low-cost (and supposedly low-performance) disk storage. In 1986, I purchased an Apple MacPlus, possibly the first PC with a small computer systems interface (SCSI) connector, along with an early Apple 10-MByte Shoebox drive. It was clear that small form-factor disk drives were for real. This got me thinking that about how we could gang together multiple drives to achieve the sort of high-aggregate I/O rates we needed to break the log write bottleneck. The idea of making use of ''just a bunch of disks'' (now commonly known as JBODs) was clear at this early stage, but I had not as yet considered the importance of reliability or how to introduce redundancy to achieve it.

As a follow on to a project Patterson had initiated to build an integrated multiprocessor architecture called symbolic processing using RISCs (SPUR),[3] he, Stonebraker, and I began a new project to explore how to support an extensible database system on top of a ''shared disk'' multiprocessor architecture. This became the XPRS Project, supported under a new experimental systems program at the US National Science Foundation. We wrote our proposal in 1986, with funding commencing in 1987. The proposal had nothing to say about adding redundancy to the disk system because we hadn't considered those ideas yet.

During early the summer of 1987, Patterson and I began seriously discussing the architectural challenges of building large-scale multidisk systems, based on small form-factor disk drives. We finally began to think about reliability; having so many more of the small form-factor drives would likely result in numerous failures. Garth Gibson, one our graduate students, became our expert on the performance-reliability tradeoffs in disk systems. This was the topic of his doctoral dissertation, which was to share second place in the 1991 ACM Dissertation Award competition.

In late summer 1987, Patterson and Gibson attended a short course on disk technology that Al Hoagland had organized at Santa Clara University. This was our first introduction to Al, who was a legend in the disk storage community. He had been a major contributor, along with Douglas Engelbart, to the California Digital Computer (CALDIC) Project at Berkeley in 1951, receiving his PhD and joining IBM in San Jose to work on the RAMAC, the project that built the world's first disk drive in 1956. Up until that time, the primary technology for secondary storage was magnetic drum, as pioneered in CALDIC. Little did we know that we had now made a connection with a man at the very center of the disk storage industry.

Our RAID prototype offered an early demonstration of many of the features that we now consider essential for modern storage systems.

During the fall of 1987, Patterson, Gibson, and I began to develop the RAID taxonomy. We came up with the levels as a way to distinguish among the trade-offs in redundancy, performance, and reliability. While working to determine the right reliability and performance metrics, we became aware of several commercial examples of redundant and arrayed storage. Tandem Computers, a company focused on high-availability systems, had long included mirrored storage in its Non-Stop Architecture. We included this in our taxonomy as RAID 1. It uses twice as much capacity as a conventional disk but can also executive twice as many simultaneous reads (since a write must go to both disks in the pair, the write performance is roughly the same as a single disk). Patterson had done consulting work for Thinking Machines, an early developer of massively parallel high-performance computers. Their new storage system, the Data Vault, was organized like a memory chip array because their processing cluster was organized as a bit-oriented single instruction, multiple data (SIMD) system. We called this RAID 2. It had less redundancy information than mirrors, but it could only perform as many reads and writes as a single disk. We also learned about announced products from the storage companies Maxstor and Micropolis that offered a system that looked like a single logical disk but were internally organized as multiple data disks with a ''plus one'' disk to hold parity. These we called RAID 3. They had even less redundancy overhead than level 2, but the same problem with limits on the number of reads and writes. Leveraging ideas on data interleaving and declustering developed for supercomputers and database machines, we developed a RAID 4 that shuffled data at the block level among data drives with parity on a dedicated drive. This organization allows the number of reads to scale up but the number of writes is still limited to that of a single disk. This led us to RAID 5, which interleaved the parity blocks along with the data. At the time we were writing the technical report, we actually thought we had invented levels 4 and 5 because we were unaware of any products that embodied these forms of RAID.

During November 1987, we sent our ''Case for RAID'' technical report to several reviewers to get feedback before submitting it for conference publication. One reviewer was Hoagland, who promptly copied it and sent it to his many contacts in the storage industry. It rapidly became the ''tech report heard 'round the world.'' We were soon contacted from all corners of the storage industry with requests for visits to our group and for technical discussions. One contact told us that he had received a dozen copies of the report within a few days, all from different sources within his company! Many told us that they had had similar thoughts about the inevitability of arrays, but our report was the catalyst that brought it all together. The paper was published as a Computer Science Technical Report in December 1987. We submitted it to the ACM Sigmod Conference in January 1988—it had nothing particularly to do with database systems, but this was the closest publication opportunity. It was accepted, but not without considerable criticism from the database systems reviewers. Nevertheless, we were able to address these, and the paper ''officially'' appeared in May 1988.

Although the Sigmod conference was our earliest academic publication on RAID, the original technical report and its informal distribution through the industry ''social network'' clearly had a much greater impact on storage system design and implementation. I have been told that one of the first EISA controller boards to be designed was a RAID controller at Compaq (EISA was announced in late 1988). Many start-ups used the RAID paper as a basis of their business plans, including such highly successful storage systems companies as NetApp. RAID was brought into EMC through its acquisition of Data General, which had the Clariion Disk Array product that was motivated in part by our paper.

After the conference paper appeared, we became aware of a rich patent literature about arrays. During a talk I gave at IBM's Research Laboratory in San Jose in the fall of 1988, the respected database researcher Bruce Lindsay told me about Ouchi's May 1978 patent, which disclosed a RAID 3 and possibly a level 4.[4] This work was undertaken at the IBM Palo Alto Science Center in the 1970s, when IBM was greatly concerned about reliability issues in the new fixed-head Winchester disk technology. The technology exceeded its reliability goals, and the array concept was abandoned. Lindsay also pointed me to a recently issued patent that disclosed RAID 5 as practiced in IBM's S/38 and AS/400 midrange systems.[5]

From 1990 to 1993, with support from DARPA and NASA, we constructed a 192-disk RAID prototype, configured to support network attachment in support of high-performance computing. This prototype now sits in the Computer History Museum in California (see Figure 2 for its predecessor's first prototype, still at Berkeley). To overcome the write overhead associated with RAID 5, we took advantage of John Ousterhout and Mendel Rosenblum's implementation of a write-optimized log structured file system.[7] Mendel, one of the founders of VmWare, won the 1992 ACM Dissertation Award for this work.

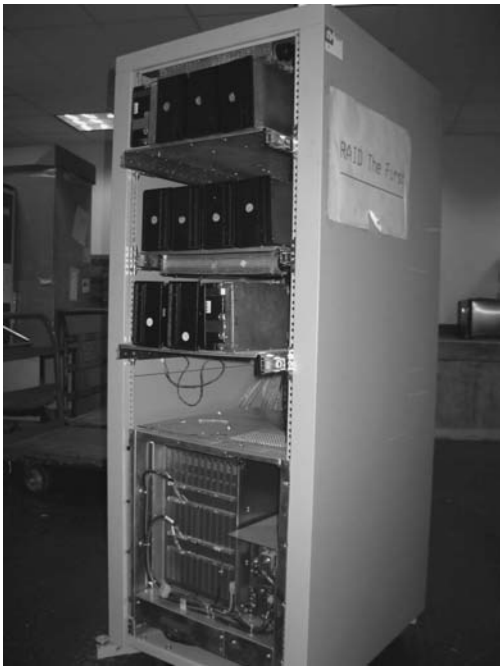


Figure 2. First-generation Berkeley RAID prototype, with RAID 5 functionality implemented across 32 disk drives. A larger second-generation prototype with 192 disk drives is on display at the Computer History Museum.

Storage as a system

Our RAID prototype offered an early demonstration of many of the features that we now consider essential for modern storage systems, particularly the mappings between logical and physical blocks, configurable redundancy support, embedded caches to accelerate performance and assist in the parity computations, and making the system a first-class network node. The prototype also supported hot replacement of failed disks, spares, and background copies and backup. Embedded in the system was a ''write-optimized'' log-structured file implementation. Today's systems combine these features with compression and encryption on the fly. Implementing RAID is more of a software than hardware design effort today, and this proved to be a big development challenge for many RAID storage companies with a stronger hardware background!

RAID Legacy

My work on RAID is probably the most enduring research contribution of my career. Patterson, Gibson and I shared the IEEE Reynold B. Johnson Information Storage Systems Award in 1999, the first systems architects to be so recognized. RAID is now a common term associated with storage systems, known to many computer users, although few are likely aware of the origin of the term stretching back 20 years to our group at Berkeley.

If you google ''redundant arrays of independent disks,'' you get 892,000 hits. (You get 410,000 if you substitute inexpensive for independent—the marketers have clearly won!) The original 1988 paper provided a foundation for much academic research in storage systems. It has more than 2,100 Google Scholar citations and won a 10 year ''Test of Time'' award from ACM Sigmod in 1998.

When I last saw reliable market research data a few years ago, the RAID market was $25 billion per year in 2002, with more than $150 billion in RAID storage device sold since 1990. There were more than 200 RAID companies at the peak. The National Academy includes RAID among the technologies created by federally funded research in universities that have led to multibillion dollar industries (also known as ''the tire tracks diagrams'').[7] Today, software implemented RAID is a standard component of modern operating systems.

RAID glossaries usually define our contribution as such:

> Berkeley RAID Levels: A family of disk array protection and mapping techniques described by Garth Gibson, Randy Katz, and David Patterson in papers written while they were performing research into I/O systems at the University of California at Berkeley. There are six Berkeley RAID levels, usually referred to as RAID Level 0 through RAID Level 5 (http://www.newyorkdatarecovery.com/raid-glossary.html).

But perhaps our most enduring contribution is our experience demonstrating how a common intellectual framework and terminology, developed by researchers outside of the pressures and positioning of the marketplace, can allow engineers and technical developers to talk with each other, exchange ideas, and ultimately accelerate the development of what became a multibillion dollar industry sector.

References

1. D.A. Patterson, G. Gibson, and R.H. Katz, ''A Case for Redundant Arrays of Inexpensive Disks (RAID),'' tech. report, CS Division, Univ. of California Berkley, 1987.
2. D.J. DeWitt et al., ''Implementation Techniques for Main Memory Databases,'' *Proc. ACM Sigmod Int'l Conf. Management of Data,* ACM Press, 1984, pp. 1–8.
3. M.D. Hill et al., ''Design Decisions in SPUR: A VLSI Multiprocessor Workstation,'' *Computer,* vol. 19, no. 11, 1986, pp. 8–22.
4. N.K. Ouchi, ''System for Recovering Data Stored in Failed Memory Unit,'' US patent 4092732, to IBM, Patent and Trademark Office, May 1978.
5. B.E. Clark et al., ''Parity Spreading to Enhance Storage Access,'' US patent 4761785, to IBM, Patent and Trademark Office, Aug. 1988.
6. M. Rosenblum and J.K. Ousterhout, ''The LFS Storage Manager,'' *Proc. 1990 Summer Usenix Conf.,* Usenix Assoc., 1990, pp. 315–324.
7. ''Funding a Revolution: Government Support for Computing Research,'' 1999; http://www.nap.edu/catalog.php?record_id=6323.

Randy H. Katz *is the United Microelectronics Corporation Distinguished Professor in Electrical Engineering and Computer Science at the University of California, Berkeley. He is a fellow of the ACM and IEEE and a member of the National Academy of Engineering and the American Academy of Arts and Sciences. Contact him at randy@eecs.Berkeley.edu.*

Contact department editor David Walden at annals-anecdotes@computer.org.

cn *Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.*