



EXECUTIVE BIO

INNOVATING@SUN

Sun Storage J4000 – Part 1

Announcer You're listening to the Sun Microsystems Podcast Network. Welcome to another edition of Innovating@Sun with your host Hal Stern. Today's topic: The J4000 product line, part one. And now, here's Hal Stern.

HAL STERN:

Hello and welcome to another edition of Innovating@Sun. I'm your host Hal Stern, Vice President of Global Systems Engineering, and I'm joined by Bill Moore, who is the Chief Engineer of our Storage Systems Group, and we're going to talk about a bunch of disks. So, Bill, welcome to the show.

BILL MOORE:

Thanks, Hal.

HAL:

And we've just introduced our new J4000 series of products where the J stands for the same thing it does in JBODs, "Just a Bunch of Disks". What's new and exciting in an idea I think that's been around the industry for a little while?

BILL:

Well, Hal, the reason is because if you look at what's been happening in the server world, is that you keep getting more and more compute and more and more I/O capability in every single server you buy. You know, whether you want it or not, that's just where the general purpose economics are driving us. So I think what you'll start seeing, and you know, of course, Sun is a big believer in this, that the value in your storage system will start moving to general purpose software running on top of industry-standard architecture and JBODs, so instead of putting all your data protection, all of your data reliability requirements off on an array somewhere far away that's built on top of proprietary margin-enhanced hardware, you'll start seeing that the true value is in the software, software like ZFS and Solaris, that will be able to take advantage of the capabilities of the general purpose compute and make what people used to think as "oh, yawn, another JBOD" into something that would be even more powerful and more reliable than your typical storage array is today.

HAL:

All right. So let's actually break that into a couple of points. The first thing you mentioned is this refactoring of the elements of the storage system into those things that represent the data management storage control, if you will, and then those things actually go and involve putting the bits onto the spinning REST, if I could be pejorative about your storage products for a moment. So I would say, let's start at the bottom and work our way up. What's new and interesting around the J4000 series? Again, JBOD products in terms of packing disks into a rack that might have been around for a while, what have we done here that's interesting and differentiating from a hardware perspective?

BILL:

So, of course, one of the things that we wanted to accomplish with this is because we're coupling them with our industry-standard servers, we really wanted to drive reliability and cost of these JBODs to sort of reflect what you see in your general hardware line overall. So, as a result, we've, I think, done a pretty good job of engineering these things such that they're reasonably priced to build and that they're reliable enough to meet the needs of the kinds of customers who buy products from Sun; that is, enterprise customers who expect a lot of reliability for the hardware that they're going to buy. Which again, is sort of a shift from what you see the typical JBODs in the industry being focused on, which is the really small to medium businesses that just don't have any money and are, therefore, are just looking for the cheapest

solution, not necessarily something that an enterprise could put their business on.

HAL:

And I guess that coupled with shifting some of the responsibility for the data protection into the general-purpose software layer reminds me in some ways of things we did a while ago with the X4500 or the Thumper product. That's something we've talked about previously on this show. Lessons learned there? Things that we learned that we wanted to reapply in the JBOD series?

BILL:

Yeah, and that is that bandwidth is, of course, a very, very important factor in all this, so that's why you'll see on the J4000 series of JBODs, they're not fiber-channel attached, and thank God they're not parallel SCSI attached. But they're attached with, well, in the storage industry, something relatively new called SAS which stands for serial-attached SCSI. Now, the interesting thing about SAS is that it has the same sort of point-to-point switch topology that a lot of your advanced SANs have today which gets you reliability, gets you concurrent throughput to the fabric but also at an economic price point because it's based on copper connectors, not fiber and things of that nature, as you get much higher bandwidth and much lower cost per port of connectivity into your disks. And as a result, you get a huge amount of performance at a much more moderate cost than if you were going with a traditional fiber-channel product. So, for example, if you look at the back of a fiber-channel product, you have a fiber channel port that runs at 4 gigabits now, transitioning to 8 gigabits. Compare that with a SAS product like our JBODs, and each port there is what's called a 4X; in other words, it's four 3-gigabit SAS channels bonded together which is 12 gigabit. That's today's speed, and early in the next calendar year, you'll see us as well as the rest of the industry transitioning to 6 gigabit per channel, which means one cable coming out of the back will give you 24 gigabits worth of throughput coming out of the box. So just like the Thumper or the X4500 that you mentioned earlier, which has tons of throughput and connectivity to the disks inside, is that's the same sort of thing we're looking to provide here, is enough bandwidth and a low latency at a low price point that you can actually keep the CPUs and your servers fed with data fast enough to operate your business.

HAL:

Well, thank you for really giving an excellent short-term roadmap for the networking side of this and how we're going to actually go talk to all these disks. To me, I think the critical problem becomes how exactly are you laying the bits out on the disk? What are you doing to go from thinking about disks as an enormous pull of bytes on which you can store things into the more logical uses of how are we going to handle our unstructured data? How do we handle our structure – what's the software architecture for actually going and managing the disks, both at a management layer as well as what the programmers use, what the applications sees? How do you go talk to these things? You know, what's the file system interface, or what's the logical organization of all those bits into something that has the appropriate performance or reliability characteristics? And I know that we've been talking publicly about our open storage effort, but it seems like there's the right confluence now of the software side along with some interesting hardware that will make this attractive to developers.

BILL:

Yes, exactly. So, you know, we just described sort of the Lego bricks that we're going to build our reliable systems out of, and then everything, as you mentioned, focuses back up on the software, which in the case of Sun is Solaris, our operating system, and ZFS which is our strategic file system going forward. Now what ZFS allows you to do is when we designed ZFS, to give everyone a little bit of background now, Jeff Bonwick and I were the two team leads for the ZFS Project leading up to integration into Solaris, so I was heavily involved in most of the design of ZFS. So the thing about ZFS is that we realized that with the shift of the market to industry-standard components and, therefore, commodity price points, is that what you would get is not necessarily the big monolithic, extremely reliable storage that you're used to. But just purely for economics, you'll get stuff that, by nature, is just designed for a slightly lower edge than those devices. However, with ZFS, we took that into account knowing that every single component that we'll be plugging into this, as Jonathan says, it's either failed or about to fail. So we designed ZFS from the get go with a notion that hardware is inherently unreliable, that every single component has to be treated not as a failure, is this crazy path that, you know, happens once in a blue moon, but it's a fact of everyday life. Because if you look at a lot of our larger customers, they literally have thousands of disk drives hooked up

to a single server. Now, at that scale, it's not like a drive fails once every year or two. At that scale, you get drives failing pretty much couple a week. So failure is something that happens all the time, every day, and the software has to be designed from the beginning to treat that as the common case, not as the exceptional case, which is what we did with ZFS. If you look into the features of ZFS, you'll see that data protection is a really big focus on the features of ZFS and on the code itself. And so, again, another important feature there is that the data protection is actually occurring in the server in the same fault domain that your application is running in. Compare this with your traditional monolithic storage array where the data protection is out across a fabric from where your application generates the data. So what this means is that with ZFS, as soon as your application generates the data, it's immediately protected and shipped out to the disk that will store it. With a monolithic storage array and, you know, traditional file systems, what happens is the application generates the data. It gets shipped across a network, which thankfully is reasonably reliable, although nothing can be perfectly reliable, out to a monolithic storage array where the protection then starts happening. So in the traditional case, there's a lot of places for bad things to happen to good data, as it were.

HAL:

Well, and I think that for a long time we've always operated under these assumptions that there were certain events that were just so low priority, we could safely ignore them, and we're talking about error rates of 10 to the minus 12th or 10 to the minus 13th. If you look at the volume of data that we generate, touch, read, write, manage in the course of a day, for a large enough enterprise, you could easily be handling that many bytes. The probability that you're going to have a bit error somewhere along the way starts to approach one. Same thing with, as you mentioned, the disk level reliability. I think we've been conditioned to think about mean time between failures as you buy a disk and it's essentially good for four or five years. We don't think, about these things are probability distributions, which is if you have enough disks running, that one thing that's 9 or 10 or 14 standard deviations in terms of early failure, you're going to see it and you're going to see it, you know, in the first couple months, and that's just a function of very large numbers here times very, very small rates. Eventually these things become integer numbers of failures that we have to go worry about. So what you described is a software model. I think that modulates what we go build with hardware. We focus on the general purpose of storage software system, and again, start to build a community around that with our open storage efforts, and it allows you to build a different set of disks products much like the JBOD series we've introduced. I'd love to hear more of your thoughts on our – as we scale this up, how much does network scale thinking goes into looking at what else we have to go do from a storage system perspective? Again, I think there are certain assumptions you make in the small arena that you're safe from certain kinds of failures. You're safe from certain types of events, and they break when you start thinking network scale, thousands of devices or tens of thousands of devices. What else do you see there?

BILL:

Yeah. So it's exactly what you described is that you have to start looking at the entire problem systemically. So you mentioned before some of the error rates that you see on devices, the drive vendors will quote you 10 to the 14th or 10 to the 15th bits. That number is a probability of a failure which you think, wow, that sure is a lot of zeros. On the other hand, if you actually do the math, that's one bit for 10 to the 14th. That's 12 terabytes. Or in, you know, 10 to 15th, it's 120 terabytes which before sounded like ha-ha, that's just such an incredible amount of data. Who really cares? But now it's like, how long does it take you to read 120 terabytes worth of data doing your business? Not that much. So you've got to be prepared to handle these sorts of error rates which, as you mentioned when you go to network scale, you wind up running into these things all the time. So the other places that this has made us focus is not only on the reliability of what's actually stored on disk, but the actual transport between where the application generates its data and where the disk gets a hold of it to store it on, as you put before, spinning rust. So in the choice of a fiber channel, what you wind up happening is typically a SAN, and now the most modern SANs tend to be more of a switched apology but, you know, more traditionally, especially when you get down to the leaf devices, you still have what fiber channel started out with, which is FCALs, it's a fiber channel arbitrated loop. Which means that just like FTE before it or Token Ring is that every drive is hooked up to the next drive, hooked up to the next drive in a big loop going back to the controller. So everyone has their opportunity to scribble on the data, as it were. In a previous company that I was at that was building large, enterprise-class storage arrays, this is exactly the sort of thing we found, is that disk

drives, yeah, you have a certain probability for the failure on each individual disk, but you also had the probability that when something did go wrong, that the drive itself had an opportunity to not only mess up its own data, but to mess up all the data that was passing through it onto its buddy. So we actually designed, in this previous company, a series of protection points between every single drive so that we could isolate who it was scribbling on the data and take him off the loop and remove that chance of failure which, again, as we saw in the actual field and in practice was, as you put, when at scale, approaching a probability of one.

HAL:

Yeah. And then what you described there is another artifact of these types of failures is that it's not just one failure in an isolated case. Okay, that disk drive failed, the light turned red, pull it out, replace it, but they tend to show up sometimes several layers of abstraction removed from where the failure was. If you have a bit error in the networking that's connecting your storage devices to your host, you're only going to see that as corrupted application data. The disk itself didn't fail, just the bit got flipped along the way. You have to be able to detect that at the most abstract layer I think closest to the application where you're actually going to go suffer the consequences of that data not coming through the way that you thought it would, or at least being able to detect it. There's been an error along the way and to go correct it where possible.

BILL:

So as you have pointed out, what this means is that not only do you have to be able to have something like ZFS that fundamentally doesn't trust the underlying hardware and validates every single byte of data that it gets back, but also in order to build a reliable system is you have to integrate these same levels of protection and the ability to recognize a failed component and remove it throughout the system, not only up at the application level but at the file system level, inside the OS, across the fabric, on the disks, everywhere, top to bottom because otherwise, yeah, you notice a problem, but who's to blame? What action do you take in order to make sure it doesn't happen again?

HAL:

Okay, Bill, well, thank you for sharing your views of what innovation's happening at the disk storage level, in particular, "just a bunch of disks" leading to just a bunch of new innovation here in Sun storage products. I'm eager to hear your views on how our recent announcements with open storage as a new open software community as well our discussion publicly about how we're going to use Flash memory, fit into changing economics. So, Bill, I hope you'll come back and join me for part two of this show.

BILL:

You bet, Hal. It's just a bunch of fun.

HAL:

Great. Well, you've been listening another edition of Innovating@Sun, and I'm your host Hal Stern.

Announcer You've been listening to Innovating@Sun. Join us next time for the latest in innovation from Sun Microsystems only on the Sun Microsystems Podcast Network.

[End of audio]