Honeycomb

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, Open Sourcing Honeycomb. And now, here's Hal Stern.*

Hal:
Hello, and welcome to another edition of Innovating@Sun. I'm your host, Hal Stern, Vice President of Global Systems Engineering. And joining me today is Josh Dobies, Software Engineering Manager for the Sun StorageTEK 5800 System, something we refer to internally as the Honeycomb Project. So Josh, welcome to the show.

Josh:
Thanks for having me, Hal. Glad to be here.

Hal:
So, I have to run with the obvious pun. In addition to being one of my favorite breakfast cereals growing up, Honeycomb makes you think of small cells of things packed together. So, what exactly is the StorageTEK 5800 or aka Honeycomb?

Josh:
Well, at the core of the architecture, Honeycomb boasts a distributed software technology that can turn a standard commodity hardware, and lots of it, into essentially a storage cloud that is very highly reliable, incredibly scalable, very easy to manage with an object oriented paradigm, and also searchable, giving customers the ability to store objects along with data that they can later use to help find what they're looking for across all their assets.

Hal:
So, you've hit, I would say, a couple of popular meems [?] there, talking about things that are searchable, scalable, and object oriented. I would say as we have been talking to our customers about quote/unquote "doing web 2.0" the concept of building a really big asset repository or data repository continues to come up as a central theme that you have a variety of data that doesn't really fit into a database. It doesn't fit any sort of relational calculus. You just want to be able to go find it based on some metadata. So, how does Honeycomb solve that problem?

Josh:
That's right. And the piece of data that we're talking about here is typically your fixed content data; that's what this solution is architected for. And this area is growing by leaps and bounds. And so you want a different paradigm and a different solution to address those needs. And when you're talking about very, very large archives, billions and billions of records, you need an easy way to find what you're looking for. And traditional file system technology is not just going to get you there. So, what's often done is a relational database is used to store information related to your assets that you can then use to search. And what's excellent about the Honeycomb technology is that it brings this relational database technology together with the storage so that you get all the benefits of being able to find what you're looking for, being able to customize your applications to fit with your particular asset repository, and you don't have the additional you know, cost of a separate database

license or the additional administrative overhead of a separate compute farm of databases. It's all integrated in one seamless appliance.

Hal:
And to push it out a little bit more, it's not that people want to go use a relational database and go build [?] two [unintelligible] of their tags [?] or of the whatever their scripters are using. It's that that's historically the only way you could get fast indexing. And now with the Honeycomb software, what you have is the ability to come up with a metadata search and storage mechanism that will also provide the indexing, so you can go pull the content off. I hesitate to call it content – addressable storage, because that brings up some you know, bad 1980s hardware defenses, but I think that's probably what you've built.

Josh:
Yeah, in a way. Another term is object storage. And certainly the content that we store is guaranteed to be the same, which is very important for fixed content repositories. And it's guaranteed to be the same over an enormous long term projection. And if you look at the reliability characteristics of this particular appliance, it has a dual parity encoding scheme coupled with some very advanced self-healing capabilities that will give you, essentially, an unprecedented meantime to data loss of over two million years. So, it's definitely going to keep your data and keep it safe for a long time.

Hal:
So, I guess you've turned a corner here and started talking about very long term compute platforms and as we talk about eco-computing, it's not just about power, space and energy savings, it's also about building a sustainable platform. And I think a number of people have noted that the more things we do digitally, the longer they tend to stick around just because there's no incremental cost to storing them, or so it seems. And some of these things we do want to keep for tens or hundreds of years, whether it's public laws or wills or other pieces of digital art that we'd like to archive for a very, very long time.

Josh:
Yeah. It's really amazing this idea of fixed content storage and just the growth of this area. You know, traditionally we're used to thinking of storage to store many of our works in progress, our documents or the spreadsheets that we know and love, but if you really look across the industry and even within your own applications, you'll see that most of the content is actually fixed. Everything from bank records to medical records, and even content that you're working on and is a work-in-progress, eventually becomes fixed. And this is the area that's growing the most rapidly; I think 90 percent growth year over year. And across all the digital assets within a corporation, 70 to 90 percent of them will become fixed within a short period of time.

Hal:
And in some cases it's nice just to fix a point in time because they'll always of course need to go backwards, whether it's for a snapshot or just to go establish a bag of things that represent a particular point in time for a project or for a number of people working together. So, I guess we should explore this notion of fixed assets a little bit more. I think historically people have thought of storage appliances as things that were read/write and what I'm hearing you say is that once you put things into the 5800 system, they become fixed.

Josh:
That's right.

Hal:

That it's a write once/read main.

Josh:
Exactly, yeah exactly. Yeah, otherwise known as WORM [sp?]. So, the technology does a lot of work to make sure that the assets that you've stored in the appliance will remain the same, again, through a number of advanced self-healing capabilities. It proactively looks to make sure that there has been no bit-rot, so to speak, of the content that you've stored within there. And as components fail, if you have failures of drives or server nodes, or even parts of your switching fabric, the system is able to reconstruct your data and bring it back into a full reliable state.

Hal:
And what's the programming interface? So, if I'm a developer, what do I see? What do I want to see looking at this? Or does it just, you know, look like file systems to me?

Josh:
Well, there's a number of options you have to integrate with the technology. The technology does have an API that you can use. It's a Java API or a CAPI that is very simple to write objects, read objects, add metadata, issue queries. And also Sun is participating with the Storage Networking Industry Association in their creation of a standard for object based storage, which is a relatively new paradigm, and that standard is called X-A-M or Xam in working together with other contributors to create a standard that can then be leveraged by multiple independent software vendors, OEMs, et cetera. And then finally we've partnered with other companies to provide additional gateways into the appliance. For example, Itochu [sp?] is a Fortune Global 500 Company that is a maker of a storage switch product that gives you the ability to interface with the appliance using more traditional means like the network file system protocol that we know and love or SIFs [sp?].

Hal:
So, you have the option of, according [?] to what you're [?] using some immersion standards, you have the option of treating it like a good old file system, and of course, then going and articulating out what you want the metadata tags to be. Where do you see out to the intersection of the way we're thinking now about our rich Internet data, if I could [laughs] if I can coin a phrase, assets that we've tagged, that we've put metadata on, we've added some descriptors to them, whether it's all the information about a photograph including the camera that we used and the exposure and perhaps any filters that were involved through to why not go and create a blog entry, I'm going to go tag it. I may provide some other context information for it. I may want to go include it with pointers to other threads of discussion. Where is that going from a developer point of view? I mean, how much do our developers have to worry about not just about the data items that they're getting, but sort of also all the context for the data that they're getting?

Josh:
Well, one thing that's really nice about the 5800 and the Honeycomb technology is that it's fundamentally a pluggable architecture. And we already see this with the ability to customize extendable metadata, but as you mentioned, there's lots of different types of content. And wouldn't it be nice if people could contribute actual pieces of logic or code that was running close to your data itself that was essentially extracting relevant metadata for the different types of assets that are in your repository. You mentioned pictures and you know, JPEGs and GIFs, for example, have a lot of metadata already embedded into the file itself. And wouldn't it be nice if, as you are storing a particular image into a storage platform, it automatically noticed that it was a JPEG or a GIF and extracted that type of meta data and made it available to your application for searching? And you can imagine doing that for lots of different types of assets. So, what I'm describing here is actually something that is a future

feature that we're working on as part of 5800 called Storage Beans. And this essentially provides a pluggable architecture to bring computing power right next to the storage power so that you can do interesting things like metadata extraction on the fly or possibly running background search algorithms over your data set to create new knowledge about what's in your repository. Or, for example, on the way out as you're retrieving an object, perhaps you want to store a very rich copy of a video, a very high-grained copy of a video, but on the way out you'd like to do some filtering to reduce the sampling bit rate, for example. This is something you can do when you have such an enormous amount of compute and memory power sitting right next to the media or the disk that's holding your storage. And so, I think that's where we're going to see it go. It's about combining the compute power with storage and giving the developer community the ability to plug in smarts [?] right into the storage platform.

Hal:
So, I think to really expand on that a little bit more, we need to pick up two threads. One is what the hardware platform itself looks like? And the other one is where are we going with the software architecture? So, let's start by going closer to the electrons and the oxide here. If you were to look at our current storage and systems product lines, I think you'd have things like the X2100 on one end. So, you know, a nice small rack mount server. Then you have the 4500, the Thumper, on the other end, which is a little bit of compute and an awful lot of disk packaged into a box. Where do you put the 5800 in that spectrum in terms of density of storage and density of compute in the packaging?

Josh:
Well, it fits basically kind of in the middle between those two that you mentioned. The core unit of the 5800 is a variant of the X2100 line. Instead of having just two slots for hard disks, it's expanded to include four. You have your AMD CPU and several gigabytes of memory. But then what happens is that because the software gives yourself the ability to stack these particular server units and make it look like one big storage farm. And so for example, a single, what we call cell, of the 5800 appliance can have up to 16 of these X2100 server variants, which provides essentially 32 terabytes of raw data along with you know, 16 CPUs and many gigabytes and memory. So, in a way, it's a little bit between the two because you've got a single unit that has a relatively modest amount of storage, but the software is able to make it appear like it's a very, very large and scalable storage cloud, so to speak.

Hal:
And can you chain multiple cells together? And to borrow on the Honeycomb phraseology here, into an even larger grid?

Josh:
Yeah, absolutely. In fact, I think this is one of the most compelling characteristics of the 5800 architecture is that you could start relatively small and literally seamlessly scale into the petabytes of information simply by adding additional cells horizontally next to the ones that you've already procured. And there's no additional management overhead associated with incorporating the new capacity into your architecture. It literally just appears as new terabytes on tap, so to speak and there's no volume or lun [?] management that is traditionally required with other storage appliances that we're used to today.

Hal:
So, clearly there's a lot of software at work here in terms of making new systems appear part of the cloud and doing the metadata management and as you described, in actually implementing the Storage Beans. What is happening in terms of building a develop community around the 5800 software system? Again, I'll point out the…

you're the software guy talking about something that people have historically considered a hardware product.

Josh:
Right. Well, yeah, there are certainly a number of differentiating characteristics of the 5800 and one of the ones that I'm really excited to talk about today is the fact that Sun is open sourcing all of the software technology to make it freely available to the development community at large. And this, you might say, is one of the most innovative characteristics of this technology offering, especially relative to the other ones that are out there that are still closed, that still give you vendor lock-in and give you less choice. For Sun, however, this is nothing new. This is exactly in line with its overall strategy for open sourcing and contributing to the developer community. And as you know, Hal, Sun is the number one contributor of open source software to the developer community at large. This is a first in that it's the first storage offering that is now open source. And you're right. It's absolutely true that with commodity hardware, software is what's needed to really bring the benefits to meet the problems that exist for storing large amounts of data. And so, it's really exciting that Sun has completely opened this up and invited the developer community to be an active participant in driving the technology from where it is today to where it's going to go. With an open solution, you now have the ability, not just to hope that the vendor is going to deliver you the innovations that you're looking for, but you actually have the power to contribute and participate on a very active basis; something that no other storage vendor can offer today.

Hal:
And I would say to put an exclamation point or even finer exclamation point on what you just said, we've even contributed the code out to the Fedora Commons. So, you can pick up the 5800 software, not just through Sun, but through Sourceforge.net [sp?] as well.

Josh:
That's right.

Hal:
And again, clearly trying to make this…the idea of the long term fixed storage object management, something that is OS independent…

Josh:
That's right, that's right. And you mentioned Fedora Commons, this is a great example of an open community that many of the digital libraries that are being built out of universities and government institutions are leveraging. And when Fedora Commons heard about the Honeycomb technology and knew about Sun's commitment to open source, they were very excited to create a tight integration and optimized integration between their offering and the 5800 technology stack. So, now you have essentially a content management system that's tailored for large libraries of digital assets that leverages the 5800.

Hal:
And some of our early success stories, if you will, with the 5800 have been in, I would say, that nice confluence, to make another pun, of long term asset management, long term sustainable computing in the university setting. So, I would say we've had some great successes already in the university library setting, in the university archives setting.

Josh:
That's true, that's true. For example, Oxford University is one that has explicitly stated that it's because of Sun's commitment to open source that they see that Sun just gets it. And we believe that open sourcing its software for

technology stack is a critical and essential requirement for any viable long term preservation project. Oxford gets that. Stanford University also gets that. They've been an early adopter of our technology and chose the 5800 over other offerings that they evaluated highlighting specifically its stability, especially in the presence of component failures like disks and server nodes, and also citing Sun's world class support customers know that they can get. And on top of that, we've made the software completely open; only helping increase the viability of our investment in that technology.

Hal:
I think there's an even stronger statement here which is, if you're talking about archiving or capturing data for longer than a current people generation. Forget about longer than the current technology generation. You have to worry about encumbering the way in which that data is stored, accessed, retrieved and turned back into information. And you don't want to go create a tax on future generations by signing them up for something that they're going to have to continue to license, buy, maintain or support. By putting things into an open source project, essentially, guaranteeing that the code's available, the community is available, and as long as there's interest in maintaining the software assets, the assets are there to be maintained without any encumbrance or any imposition on the people who most likely will want them at some point in the future and may or may not have any historical context for how they happen to get the assets.

Josh:
Exactly. You know, at the end of the day, the customers are essentially responsible for the assets that they're storing. And unfortunately, with a lot of the closed solutions that are out there, companies just aren't giving customers the power to actually protect their assets. And Sun gets that. That's why they're open sourcing the software technology.

Hal:
Great. So, where should people go to find out more information, get started, come up to speed on the Honeycomb technology?

Josh:
Excellent. I'm glad to hear that question because with today's announcement, obviously, the call to action is clear and very loud and that is to start participating. The code is available. It's open. Freely available for download. I imagine people would be able to see some links next to this link for the podcast where they can get more information about the StorageTEK 5800. If you simply go to Sun.com and search 5800 in the search box, you will find the product page. One thing you can do that's really exciting and gives you the ability to start using this technology immediately is downloading the software development kit that comes with the 5800 system. This gives you the ability to start playing with the API interfaces and seeing this new paradigm for dealing with object based storage and using metadata and search to integrate with your applications. So, whether you're an independent software vendor or a potential OEM partner or a direct customer, an IT manager, or even a hobbyist, we're calling on all of you to start participating with Sun through the openness of the 5800 technology stack. You can see this on Opensolaris.org as well as Java.net.

Hal:
Great. Well, certainly exciting stuff and it's great to see us tying together a number of the themes we've been discussing, I'd say, over the last 18 months to two years, from emphasis on storage built out of general purpose systems to how open source software fits in here to, you know, really thinking about storage as a software problem and continue to go build developer communities. So, Josh, I'd like to thank you for joining us on the show and giving us something of where we are with Honeycomb technologies and you've been listening to

another episode of Innovating@Sun. 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.