Wednesday, September 2. 2009

Sun Storage F5100 Flash Array

Sun talked about a flash array at the ISC in Hamburg. Now it got obvious, that something is imminent in regard of this system, as soon as patch 141484-01 was released a few days ago. Yesterday the news broke about some documents at docs.sun.com about an interesting upcoming device. Now Storagemojo wrote about it Kudos to the Sun engineers who have been driving flash forward faster than almost anyone else in the industry. as well as Octave Orgeron in his blog who statedThis will be Sun's first step into selling products that could very well change the storage landscape.Perhaps those was fueled by this teaser blog entry yesterday, but there was an other article about it at blogs.sun.com. Looks i'm not the only one traversing docs.sun.com at a regular schedule.

This device is really something really interesting, but i won't write more about it until it's officially announced. Oh ... just one teaser question: What's the really slow and performance-annihilating part at swapping virtual memory from/to disk?

Posted by Joerg Moellenkamp in English, Oracle, Solaris, Technology, The IT Business at 20:22

Preis? Ist bestimmt nicht billig.
Anonymous on Sep 2 2009, 22:09

Habe doch am Ende des Texts geschrieben, das ich mich zum Announcement zu nichts zusätzlich aeussere
Anonymous on Sep 3 2009, 09:22

Hmm - aber SSDs für eine X4540 sind immer noch nicht lieferbar
Anonymous on Sep 3 2009, 10:45

"Oh ... just one teaser question: What's the really slow and performance-annihilating part at swapping virtual memory from/to disk?"

Wow! Didn't you just a few days ago dismiss the idea of putting swap on a SAN because it was too costly? Now you suddenly think it is smart putting swap on an even more expensive flash storage array. I mean, show some consistence here.

I have a quiz too.

If you have a thrashing system what will you do to try to restore performance.
a) Switch to faster disks for swap
b) Eliminate thrashing. Analyze the system, tune application/OS and/or add memory.

b) will get you the sysadmin job. If you answer a) you are barking up the wrong tree.
Anonymous on Sep 3 2009, 12:44

1. I knew it from the moment i've posted this article that you would write such an comment ... you starting to get predictable.

2. I knew it from the moment, i've posted this article, that you wouln't understand where this teaser question is headed too.

3. I will help you with a additional teaser question: "What kind of system would you need to provide an application 4 TB of memory and how much would it costs?" You are allowed to answer in IBM pSeries systems
Anonymous on Sep 3 2009, 13:17

A thrashing system will reduce the overall performance to a fraction. Even if the swap space resides in the cache of an USP-V(which is way faster than your flash array) the performance still sucks(I have tried). So everybody wants to avoid it. But you want to actually have a system that is designed to be thrashing all the time.

So you have this big application that is really, really important. It runs on costly hardware and with a costly infrastructure. Oracle licenses may already have set you back $5m. So instead of shelling out the bucks needed for the 4TB of memory and get the maximum performance you let Oracle have its running processes on flash disks connected with a SAS cable and obtaining 1% of the performance.

I think the word describing this is "pennywise".
Anonymous on Sep 3 2009, 17:18

Joerg,

It turns out that, unbeknownst to me, the person who linked me to the Sun docs had seen them on your blog.

Keep up the good work.

Robin
Anonymous on Sep 3 2009, 19:01

0. At first ... why do you start to insult me? At first with this "admin job" thing and at second with this "pennywise"?

1. I'm wondering why a technological guy shows such a lack of imagination and innovation. Of course you won't substitute something like a 4 TB oracle database on a M9000 or on a p590 with a X4170 and this F5100 with a database that really fills the RAM with 4 TB worth of hot data.

2. Believe it or not, there are workloads (and they are not rare), that doesn't need much compute power, but react really grateful performancewise on the existence of large memory areas. The problem: Often you have equip those large memory machines with vastly more CPUs than you need, sometimes just to provide the sockets for all the memory (you know ... 4 TB need a lot of DIMM slots) or because there are some side conditions that mandate some CPU on a board with memory.

3. Now assume this use-case: A customer has a multi-terabyte set of data. It's a assumption out of experience that some of this data is really hot, some is rather warm, and some data is seldomly used. This is the same assumption that led to L2ARC and there are several examples that this stuff works. Even without zfs there are good examples for the capabilities of an SSD. When they used an database on this ssd there were able to reach almost the performance of an memory saturated configuration of the database (http://www.c0t0d0s0.org/archives/4990-Mysql-on-SSD-benchmarked-....html). On Opensolaris i would obviously use L2ARC, but there isn't L2ARC on Linux for example and the current version of Solaris doesn't have L2ARC as well.

I think we both can say out of experiences, that caches work. I know from personal experiences that the L2ARC concept works. You will find other resources in the web that suggest the same. And now we need a transfer job in the brain: What's different, when we doesn't use the virtual file system layer as the API for the SSD and use the virtual memory layer? Using memory is something every application can do ... on any operating system. Using virtual memory is something every application can do on an operating system capable of providing virtual memory. So we just use one of the simplest ways to manage data.

Okay, let's assume that you have to do a big conversion job, you have a large lookup table or you play with otherwise large data sets. Now the F5100 comes into play. I'm disclosing nothing when i tell here, that this little device delivers over 1 million IOPS. When you read the documentation, you will see, that you don't connect 1 meager SATA cable to it, you would use several controllers to it connecting the device with your system.

Look into the documentation, that was published a few days ago: You can connect up to 16 HBA to the F5100 via mini-SAS 4x, thus you could connect the system with up to 16*4*3 GBit/s yielding 192 GBit/s maximum throughput on the SAS lines. Of course the latency isn't up to the access speed of RAM, but it's some orders of magnitude faster than a disk. Look for example to an M5000. It's only capable to hold 256 GByte of memory, but you could augument the system with 4 TB of flash memory, albeit just connected with 10 HBAs There are some reasons why the absolute-max-perf config needs 16 HBA

Of course you will find workloads that bring such a configuration to it's knees for example when your memory isn't large enough to hold just the hot data. But let's assume a more realistic workload, then this configuration should give you a nice performance boost. Of course not at 100% of an configuration with pure RAM, but vastly above the 1% you try to postulate here.

The idea behind the idea of using the F5100 (or any other high speed SSD) as swap is simple: What do you do, when your filesystem doesn't have such niceties like the l2arc but the complete dataset doesn't fit in your RAM?

And it helps you to shorten the run time of processes without the hassles of data management for the SSD with the interface the VM instead of being hidden in a filesystem.

At the end it's one decision the admin and his management has to make: Invest many dollar bills in large machines for top notch performance. Or using smaller system, but waiting for a long times for results. Or a solution between those extremes. Of course this has to be done after an exact analysis of your workload, but i may keep you from buying a new machine when your system is already maxed out memorywise.

4. Regarding your system that got dog-slow after using the cache of your USP-V as swap. Did it came to your mind that you were the victim of an effect we talked a few weeks ago called latency. Did you connected your test system directly to the USP-V via a short cable or were some longer cables, a switch or a director between your system and the USP-V. A Brocade 48000 introduces a port to port latency of 3,6 microseconds alone and that's a director with the reputation of being one of the fastest AFAIK. Of course a smaller switch whould have a lower latency round about 700ns for the 5100, but i have some doubts that someone who uses a USP-V, has some small 5100 in front of it

Well ... with this test you somewhat proofed my comment, that SSD shouldn't be placed in the SAN, at least for read-caching. Those latencies weren't a problem when our disk just had a IOPS rate of 100 or 200 IOPS. At over 1 million IOPS it's a completly different story.

6. It doesn't matter if the cache of the USP-V is much faster than the F5100 IOPS-wise. You have a massive bottleneck between your system and the cache. As you didn't described you implementation, i have to assume, that you've used a system with a single SAN connection. A single HBA from Emulex for example delivers round-about 150.000 IOPS at a rather low block size. An USP-V is rated at up to 4 Million IOPS. Now guess, how many IOPS you get for the system you've attached with one FC adapter. Yes ... 150.000 IOPS. A SAS controller plays in the same ballpark IOPS wise but doesn't suffer under the prolonged latencies introduces by all the gear behind the connector.

5. Sorry, when i have to insult you now: Your comment clearly show that you didn't worked into the available data and just saw a good opportunity for attacking me. Furthermore your thinking seems to show a severe lack of innovation and creativity in the light of new technologies. Oh, that wasn't insulting ... of course not ... i don't to lower myself to your level by firing back on such insults.
Anonymous on Sep 3 2009, 22:11

0. Thanks for the long and informative answer. I take the point and will stay away from (most of) the sarcasms.

1. Apart from the imagination thing at least we agree on how to meet the memory requirements on the p595/M9000.

2. Here I have some difficulty seeing the market for servers with very little CPU and massive memory that you describe. This flash array seems to target very marginal needs. I am hugely in favour of generic solutions that solves most problems(within a SAN environment for example). And solutions for exceptional storage needs can hardly be described as revolutionary for the industry. And wasn't it the 7000-series that was supposed to revolutionize the storage industry? Kudos to SUN for managing several revolutions at once.(Sorry, could not resist.)

3. This part is interesting. You name some different approaches to handling hot data. One is to cache a lot in RAM which is expensive. Another is not cache a lot in RAM and have ZFS which with a combination of SSD and spinning disk can do the tiering for you. But with hot data in VM populating both RAM and relatively much slower flash I see some potential problems. First the OS must be rewritten to treat swap space in a new way or else it will go bananas transporting pages between RAM and swap while simultaneously using a lot of CPU scanning RAM looking for candidate pages to swap out. Then you need some kind of second level swap space to take care of one of the functions swap is meant for in the first place, namely hold the data for sleeping processes. I think this approach of active swap space may reveal some practical problems.

The tiering within ZFS is a good idea but as you say this method won't work in Linux for example. Then we get to technology which I really think will revolutionize the storage industry and that is doing the tiering in the storage system itself. Up till now automatic tiering has, with one small exception(Compellent? dont remember), been on the LUN level and a little cumbersome to implement(USP). But in a year or two block level tiering or sub-LUN tiering will be commonplace with probably EMC first. So with RAM and SSD at the top tiers and SAS and SATA at the lower tiers your hot blocks and cold blocks will move around based on policy settings and access patterns. I believe this is the future of storage. Simple(no need to statically allocate storage based on assumptions on performance needs) and generic(doesn't matter which OS, application or file system is on the client side).

4. It was DR testing, hence the little memory on the server and the switch instead of director.

6. If 150 000 IOPS isn't enough, attach more HBAs. We've been through this before and I think the latencies of SAN(a million IOPS per port as a maximum) is purely theoretical. Show me the server that can do 4 million IOPS and then we can compare your SAS-connected storage with FC connected storage.
Anonymous on Sep 4 2009, 01:23