

Saturday, March 22. 2008

The datacenter of tomorrow

(this is a translation of this article: Das Rechenzentrum von Morgen)

There was an article in about an presentation of two colleagues: GUUG-Fachgespräch: Wie Sun die RZ-Architektur von morgen sieht ("GUUG Fachgespräch: How Sun views the datacenter of tomorrow"). It's a not really a positive one, but you can't expect something different from a magazine, that carries the name of the penguin operating environment in it's title. At other points the article ist just downright false: The colleagues from the Project Indiana would thank you, if you tell them, that Project Indiana ist just about the integration of xVM in Opensolaris. BTW: The invested worktime of this project so far would be really high, if that would be the case ..

Well, i don't like comment on such articles. But i know this project very well. Out of reasons i won't comment here. Some ideas are mine, others were heavily influenced by me. Nevertheless i want to say a thing or two

Well, the commentator in the Article is correct: The architecture isn't rocket science. It's not the datacenter of tomorrow, it's more like the datacenter of this afternoon (i wrote the original text in the morning). The datacenter of tomorrow would use a different interconnect, Infiniband for example. And when you look at the development in Solaris, you will see that a lot of stuff happens in the Infiniband sector. The datacenter of tomorrow will be a single fabric one. Will it be IB or 10 GbE? The future will tell. We will see SRP over IB or SRP over iWARP over 10GB instead of FC. NFS won't be a TCP/IP based protocol. We talk about file sharing via NFS over RDMA over IB.

But: The Datacenter of this afternoon is a nescessary step inbetween. To use the datacenter of tomorrow you should wait until tomorrow. Okay, you could use it today, whe you want wear asbestos longjohns in escalation meeting. You cant explain every customer, that you have to BFU an important feature into the kernel ... so ... please wait at least until this evening.

The point: From the view of the operational concepts you won't see much change because of the new interconnect. It's not relevant if the datacenter of this afternoon uses ethernet and the datacenter of tomorrow uses IB. The concepts stay the same, thus is concept presented by Matthias and Tobias are the datacenter of tomorrow .

I hadn't the time time to look in the final presentation of Tobias and Matthias, but the point of the architecture wasn't the usage of NFS or iSCSI for diskless servers. It was a different view to virtualisation. Many concepts look at systems at whole and to move systems as whole. But that isn't sufficient. I talked before about my opinion that virtualisation doesn't solfe a problem, at least none relevant in the world of UNIX. On an actual unixoid operating system you have only few reasons to use virtualisation (at least the hypervisor based flavour). Unix was virtualisation right from start and modern developments gave Unix the capabilities to control the competing processes (i hope you've read my Solaris Resource Management tutorial). There is a single wish that you can't fullfil with standard Unix: Live Migration. I know, IBM will tell you they can to it with AIX6 but that's another topic for long explanation and a higher blood pressure). Hypervisor based virtualsation can do something like that and promise short failover times. Live Migration, vmotion or partition mobility ... the tools of the assumed administrative paradise.

The problem ist: Every new technology generates new possibilities but new challenges as well. And it generates classes of problems you didn't thought of before. Swapping the metal beneath the operating system and the os doesn't notice. Okay, the operating system may notice nothing, but that is the end of the road. The promised short failover isn't possible with live migration as well, at least when you have an application that does real work on it's memory pages. There are pathological cases almost unlivemigratable but at least you have only short migration time in most of the cases. But short means seconds not milliseconds. And this opens a whole new can of problems. Let's assume the non-live part takes 5 seconds. In this time the clock of the migrated system doesn't run. You can choose two between two deaths: A incorrect clock or you can set the clock. But then you doesn't have an continious time. At least with application in need of a precise time this can lead to interesting effects. There are several reasons, why xntp doesn't set the time in one large step but makes seconds longer or shorter to drift the time to the correct time over a longer persiot. Live Migration generates time differences in the multi second range. Simply ignoring it doesn't solve the problem. There are workarounds for this problems and some are already implemented in products, but this workarounds aren't perfect at all.

But i think Live Migration is a good technolgy. xVM and LDOMs will support that. When i have to decide between an incorrect time or "system down" because of a dying processor fan, i would opt for the second possibility. But it doesn't

have to be this way: I know customers that opt for a failing system to prevent the problem of an noncontinuous time in their system)

Another solution, dismissed as low-tech, doesn't know this problem: It's a solution that many people put into the web services realm: Loadbalancers. At most time people answer: But i haven't 30 webservers ... what should i do with loadbalancers: A real subsecond failover. Assume the following scenario: I have a service in my network, that runs only on a single server. Now thing about the trick, that the user doesn't access the real IP of the service ... they use a virtual IP provided by a loadbalancer. When i want to migrate on a new system, i just have to change the configuration of the loadbalancer to redirect the requests from the old server to the new server. You can do this as a hard cut or by a slow migration by redirecting only new session. The positive point: The systems are seperated. The clocks of the systems running continious. No hypervisor that eats away performance. This method is a viable solution for many tasks. When i think about customer installation and what services i found in VM, i have my doubt, if this concepts are really thought to an end (Youknowthatimeanyou: "Yes, i'm still not convinced, that an mailserver has to run in a VM")

I know. This isn't a solution for every problem. But i wrote it in a different article. There is no universal solution. You have to look for the most adaequat soltion. You won't use a VMware with Live Migration, when i can't afford the failover time of a normal cluster. You would use RAC (bullshit for scaling, but great for accelerating failover times). An internet server doesn't need VM, they simply need Unix. Desktop Virtualisation ... okay ... VM make sense there. By the way: Those presentations of Live Migration by showing an uninterupted Video while migration show nothing. Any decent video player buffers several seconds worth of video. More than enough to complete an live migration.

I know: Many admins and decision makers (especially the later ones, as analysts and consultant prayed for cost reduction by standardisation ... come hell or high water) dislike solution you can't use uniformly in their datacenter. The question is, were you start this standardisation. I assume, you don't want a singular virtualsation technology, you want a singlar administration of all your virtualisation technology. You don't want a singular virtualisation technology that is great for some workloads, mediocre for the most and contraproductive for others. Initivatives like libvirt or Sun xVM Opcenter try to solve this challenge. We have to wait for the final outcome of this initiatives.

A part of the "... of tomorrow" lies in the way, how we orchestrate all this components. How we make a whole out of it. Architecture is more than metal, more than cables, more than silicon.

But i assume, this will be a hard birth. One example: VMware recognized, that the hypervisor alone will bring them nowhere in the future. There economic prosperity lies in the administration interfaces. The unique selling point of those interfaces lies in the capability to control vmware. Why should they give up this advantage. Well ... interesting times ahead of us.

By the way: My candidate of the dominating virtualisation solution of the year 2013 are the matured xen-based hypervisor solutions. But not in the form of the solutions known today. My prediction: One of the big BIOS developers for x86 will take an xen-enabled operating system, reduce it to the max and make it a part of their bios. I don't what implementation this will be, but i assume this developer will use an implementation that supports the economical usage of their work at best.

To get to the point of critic regard in Opensolaris at end: The point, that the hypervisor is an Opensolaris without patches is irelevant. You can look at the hypervisor as a super-BIOS, you swap as whole. It would be most sensible to deliver it as a kind of live USB-Stick you put into the USB-Ports.

As you use ZFS for your disk, you could use a filesystem localdisks/config, where the system stores it's configuration. Updating would be limited to swapping the USB-Port. Would be much simpler, you wouldn't patch a BIOS either. You have to say goodbye to the notion, that you have to use the hyperhivisor like Solaris, just because it is a Solaris. I know ... this is hard for die-hard Solaris Admins, but hey ... i have to get used to the fact, that we like x86, Scotty doesn't make jokes about Microsoft and that Apple doesn't use PowerPC for their computers