

Re: Did someone ask HP for VAX to Itanium cluster possibilities?

Source: <http://unix.derkeiler.com/Newsgroups/comp.os.vms/2003-10/2038.html>

From: Keith Parris (keithparris_NOSPAM_at_yahoo.com)

Date: 10/23/03

Date: 22 Oct 2003 23:12:20 -0700

JF Mezei <jfmezei.spamnot@istop.com> wrote in message news:<3F96F8BC.ED724D9B@istop.com>...
> *Does the clustering code have any platform specific handling ? For instance,*
> *does an Alpha treat a VAX any differently than another Alpha ?*

All nodes (whether VMS or not, e.g. something like an HSJ/HSD or even an MTI FDDI storage controller) have to conform to standards like the System Communications Architecture (SCA) and Mass Storage Control Protocol (MSCP), which define the content of data structures each implementation must include, the exact format of any messages exchanged (including field length and endianness -- Little-Endian in this case), and the allowable behavior of an implementation. These standards are OS and architecture independent. (You could use CI and HSJs with DECsystems, as I recall.)

Above these standards are what amount to standards but are more VMS-specific, and, outside of Ruth Goldenberg's and Roy Davis' books, documented mostly in the form of the VMS source listings, like the format of messages the Connection Manager or Lock Manager pass among each other and the algorithms they use.

Each node tells the other nodes what capabilities it can handle, such as VIOC and XFC, Volume Shadowing Write Bitmaps, cluster-wide logical names, class scheduling, dynamic lock remastering [(1) if at all (added in 5.2, and then allowed based on relative activity levels in 5.5) and (2) lock-tree remastering via block-data transfers instead of lots of sequenced messages], etc. and there is often special-case code present (at least over a specific period of time) to deal with a node's inability to perform a given function.

One interesting example is that the VAX can do Dynamic MSCP Server Load Balancing as that code was added in V6.0, but Alpha never got that capability, as the source code branched off as a separate stream starting at about 5.4-1, and that particular feature got left out on Alpha (although we've obviously been able to live without it). Another example of a feature that never made it to Alpha from VAX is CI Port Load Sharing. But in most cases of feature mismatch, it's the

VAX which lacks a capability added later in time on Alpha (such as mastering Write Bitmaps for Volume Shadowing Mini-Copies, or having the DCL commands and other support to dynamically expand a Files-11 volume on-the-fly).

But as you can see, there have been a lot of capabilities added to VMS clusters over the years, requiring upward- and downward-compatible modifications to the format of messages exchanged and to the algorithms within the cluster code and even various other pieces of VMS higher up. To provide upward compatibility, sometimes there are additions required in one version, in anticipation of being able to operate correctly with a node having an anticipated feature expected to be included in a future version.

- > *Or is the clustering code and protocol totally agnostic to a point where if an*
- > *PowerPC or Z80 version of VMS became available tomorrow, I could include in*
- > *in a VAX or Alpha cluster with those machines simply accepting the platform*
- > *name as a string that doesn't get used in conditional execution ?*

There are a few special cases. For example, if the software type is "VMS " instead of something like "HSJ8", in SHOW CLUSTER/CONTINUOUS we will pull the HW_TYPE field information from the more-extensive information in the Cluster System Block data structure created by the Connection Manager instead of the field in the SCA System Block (which SCA defines as only having 4 characters and thus only says terse things like "VAX " or "ALPH" or "HSJ5"), making the assumption that if the SCA software type field says "VMS " there will be a VMS\$VAXcluster SYSAP connection to that node, since all VMS nodes will presumably have a Connection Manager. (At one point in the past, there was a Standalone MSCP Server that could be built from the VMS sources, kind of like Standalone Backup was --- this would act like an HSJ and serve disks, but not participate in a cluster, just like HSJ controllers do not have a VMS\$VAXcluster SYSAP, yet communicate with MSCP protocol over SCA with VMS nodes to serve storage. So if that still existed, it would presumably have to report something other than "VMS " now.)

But to answer the question, in general if the mythical code claimed to be "VMS " in the SCA software type field upon formation of a virtual circuit, it would be treated as such, and expected to behave as such. If it did not behave properly, real VMS nodes in the cluster might crash, or other harm might occur.

- > *In other words, for an Alpha to support a new platform in a cluster, must the*
- > *Alpha have any code specific to that platform ?*

With each new platform addition, there have been some minor changes, for example, to things like F\$GETSYI for the ARCH_TYPE and ARCH_NAME item codes so the old architecture(s) know what to call the new architecture.

comp.os.vms: Re: Did someone ask HP for VAX to Itanium cluster possibilities?

- > **IF* VAX doesn't cluster well with IA64, would it be more likely because the*
- > *VAX is missing something to interface to IA64, or because IA64 forgot to*
- > *include an IF statement to deal with VAX architecture ?*

I'd guess the most likely areas would be cases where the technology or feature spread was just too great, coupled with the fact that when a combination is not tested, things could slip through.

Much time and effort over the years has gone into supporting rolling upgrades. This is not easy, as you have to have code that can act two ways (and even use different message formats) depending on what other nodes are in the cluster. So a given release has to have different code paths. Then at some point later on, once you're past the point where you need to inter-operate, you may have cause to go back in and carefully rip out the backward-compatibility code, if for reasons of maintainability if nothing else.

So I'd guess the most risky area is probably code being ripped out on Itanium as being no longer necessary to talk with Alpha, but needed to talk to VAX. Because HP isn't testing VAXes with Itanium, such as case could slip through testing without being noticed. Of course, having common code between Alpha and Itanium makes this much more unlikely, as it would presumably have to be conditionalized code for Itanium only. But you can see how it might be possible.