

## Re: [PATCH 0/7] dlm: overview

**Source:** <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-04/7154.html>

---

**From:** Daniel Phillips ([phillips\\_at\\_istop.com](mailto:phillips_at_istop.com))

**Date:** 04/28/05

To: Lars Marowsky-Bree <[lmb@suse.de](mailto:lmb@suse.de)>

Date: Thu, 28 Apr 2005 16:53:20 -0400

On Thursday 28 April 2005 10:57, Lars Marowsky-Bree wrote:

> On 2005-04-27T18:38:18, Daniel Phillips <[phillips@istop.com](mailto:phillips@istop.com)> wrote:  
> > Uuids's at this level are inherently bogus, unless of course you have  
> > more than 2\*\*32 cluster nodes. I don't know about you, but I do not have  
> > even half that many nodes over here.  
>  
> This is not quite the argument. With that argument, 16 bit would be  
> fine. And even then, I'd call you guilty of causing my lights to flicker  
> ;-)

BlueGene is pushing the 16 bit node number boundary already, 32 bits seems prudent. More is silly. Think of the node number as more like a PID than a UUID.

> The argument about UUIDs goes a bit beyond that: No admin needed to  
> assign them; they can stay the same even if clusters/clusters merge (in  
> theory); they can be used for inter-cluster addressing too, because they  
> aren't just unique within a single cluster (think clusters of clusters,  
> grids etc, whatever the topology), and finally, UUID is a big enough  
> blob to put all other identifiers in, be it a two bit node id, a  
> nodename, 32bit IPv4 address or a 128bit IPv6.  
>  
> This piece is important. It defines one of the fundamental objects in  
> the API.  
>  
> I recommend you read up on the discussions on the OCF list on this; this  
> has probably been one of the hottest arguments.

Add a translation layer if you like, and submit it in the form of a user space library service. Or have it be part of your own layer or application. There is no compelling argument for embedding such a bloated to cman proper (which is already fatter than it should be).

> "How is the kernel component  
> configured which paths/IP to use" – ie, the equivalent of `ifconfig/route`  
> for the cluster stack,

## Linux-Kernel: Re: [PATCH 0/7] dlm: overview

There is a config file in /etc and a (userspace) scheme for distributing the file around the cluster (ccs – cluster configuration system).

How the configuration gets from the config file to kernel is a mystery to me at the moment, which I will hopefully solve by reading some code later today ;–)

- > *Doing this in a wrapper is one answer – in which case we'd have a*
- > *consistent user-space API provided by shared libraries wrapping a*
- > *specific kernel component. This places the boundary in user-space.*

I believe that it is almost entirely in user space now, with the recent move of cman to user space. I have not yet seen the new code, so I don't know the details (this egg was hatched by Dave and Patrick).

- > *This seems to be a main point of contention, also applicable to the*
- > *first question about node identifiers: What does the kernel/user-space*
- > *boundary look like, and is this the one we are aiming to clarify?*

Very much so.

- > *Or do we place the boundary in user-space with a specific wrapper around*
- > *a given kernel solution.*

Yes. But let's try and have a good, durable kernel solution right from the start.

- > > *Since cman has now moved to user space, userspace does not tell the*
- > > *kernel about membership,*
- >
- > *That partial sentence already makes no sense.*

Partial?

- > *So how does the kernel*
- > *(DLM in this case) learn about whether a node is assumed to be up or*
- > *down if the membership is in user-space? Right! User-space must tell*
- > *it.*

By a message over a socket, as I said. This is a really nice property of sockets: when cman moved from kernel to user space, (g)dlm was hardly affected at all.

- > *For example, with OCFS2 (w/user-space membership, which it doesn't yet*
- > *have, but which they keep telling me is trivial to add, but we're*
- > *busying them with other work right now ;–) it is supposed to work like*
- > *this: When a membership event occurs, user-space transfers this event*
- > *to the kernel by writing to a configfs mount.*

Let me go get my airsick bag right now :–)

Let's have no magical filesystems in the core interface please. We can always add some later on top of a sane base interface, assuming somebody has too much time on their hands, and that Andrew was busy doing something else, and Linus left his taste at home that day.

- > *Likewise, the node ids and comm links the kernel DLM uses with OCFS2 are configured via that interface.*

I am looking forward to flaming that interface should it dare to rear its ugly head here :-)

- > *If we could standarize at the kernel/user-space boundary for clustering, like we do for syscalls, this would IMHO be cleaner than having user-space wrappers.*

I don't see anything wrong with wrapping a sane kernel interface with more stuff to make it more convenient. Right now, the interface is a socket and a set of messages for the socket. Pretty elegant, if you ask me.

There are bones to pick at the message syntax level of course.

- > > *Can we have a list of all the reasons that you cannot wrap your heartbeat interface around cman, please?*
- >
- > *Any API can be mapped to any other API.*

I meant sanely. Let me try again: can we have a list of all the reasons that you cannot wrap your heartbeat interface around `cman _sanely_`, please.

- > *That wasn't the question. I was aiming at the kernel/user-space boundary again.*

Me too.

- > > > *... which is why I asked the above questions: User-space needs to interface with the kernel to tell it the membership (if the membership is user-space driven), or retrieve it (if it is kernel driven).*
- > >
- > > *Passing things around via sockets is a powerful model.*
- >
- > *Passing a socket in to use for communication makes sense. "I want you to use this transport when talking to the cluster". However, that begs the question whether you're passing in a unicast peer-to-peer socket or a multicast one which reaches all of the nodes,*

It was multicast last time I looked. I heard mumblings about changing from a udp-derived protocol to a sctp-derived one, and I do not know if multicast got lost in the translation. It would be a shame if it did. Patrick?

- > *and what kind of security, ordering, and reliability guarantees that transport needs to*

> *provide.*

Security is practically nonexistent at the moment, we just keep normal users away from the socket. Ordering is provided by a barrier facility at a higher level. Delivery is guaranteed and knows about membership changes.

> > *Of course, we could always read the patches...*

>

> *Reading patches is fine for understanding syntax, and spotting some  
> nits. I find actual discussion with the developers to be invaluable to  
> figure out the semantics and the intention of the code, which takes  
> much longer to deduce from the code alone; and you know, just sometimes  
> the code doesn't actually reflect the intentions of the programmers who  
> wrote it ;-)*

Strongly agreed, and this thread is doing very well in that regard. But we really, really, need people to read the patches as well, especially people with a strong background in clustering.

Regards,

Daniel

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@vger.kernel.org

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>