

Methodics

8/16/2018

warpstor software up modifications

Goals:

- * ~~the~~ ease of upgrade
software upgrading
that is completely automated
with ~~no human~~ no or little
operator attention required
- * upgrade roll back method
- * configuration reset capability

In depth goal details and requirements.

Ease of software upgrade

Currently, upgrade of software is almost completely manual process, and not really a process at all, just ad-hoc activity and often takes a lot of time and multiple tries. The warpstor software itself is packaged, but other things like the kernel and userspace system software is not. Furthermore, warpstor config data isn't always compatible across updates, so the config data is reset when the package is updated.

All of these deficiencies would ideally be addressed in this revision.

Upgrade rollback method.

Sometimes an upgrade or software update is not satisfactory for some reason and the customer (or Methodics) would like to "go back to the way things were", in other words, roll back to before the update. And possibly to be able to roll ~~back~~ back many multiple upgrades, or even roll forward.

Access to forward or back ward "versions" w/o ~~creating~~ ^{running} them, would ~~also~~ also be very useful, both for forensic uses and for engineering uses.

Transmission of the backward versions to Methodics would also be cool. Disaster recovery would be one such use of this. Support and engineering ~~would~~ would be another. DR includes off site backup, ~~the~~ both of which could be billed separately to customers beyond usual yearly support contract.

Configuration Reset for Warpstor

Some times the configuration just needs to be reset to factory defaults. We've all had to do it to devices and computer appliances ~~to~~ before, and Warpstor is no different. Also, the factory defaults are likely to be different across ~~of~~ different versions of warpstor, possibly even for the same config item.

Does configuration reset include wiping out all ~~user~~ masters & snapshots? What ~~was~~ would config reset look like if it didn't include wiping out data sets? Two forms of config reset, one ~~complete and one~~ total and one all but datasets?

Possible solutions for Goals

* Ease of upgrade

Possibly create our own kernel and boot loader package to start. Forgive boot ~~loader~~ loader, because I think we custom modify grub. Or maybe that means we do grub ~~first~~ a custom grub package first, and a custom kernel package that depends on our custom grub package second.

Mods to the ~~warp~~ warp to package to manage ~~of~~ config file upgrades. Debian packaging already has all the pieces for this, so that can be done.

At package upgrade time the operator may have to answer some questions, so there's that. Perhaps we ~~could~~ could always engineer it not to do that and just have default behavior, but experience says that's not always possible.

cont -

The ~~ref~~ config data can be kept in ~~its~~ its own dataset, and snapshotted as necessary for roll backs. The active ~~snapshot~~ or filesystem would be ~~mounted~~ mounted into the FS hierarchy ~~so~~ so users to software would not require special knowledge of this ~~snapping~~ snapping nor would it need extensive coding to manage it. This ~~is~~ is also applicable to rollbacks and factory reset features.

Implementation details, cont.

Upgrade roll back capability

This can be implemented using snapshots of the file system and customer data sets, and the configuration configuration data.

Perhaps for the initial implementation of this feature, we should not try to snapshot the customer data ~~masters~~ (masters). In the future, the way that snapshots of ~~masters~~ and clones of masters interrelate may change in incompatible ways, so a rollback may require snapshots of those things. ~~But~~ But eventually we will have to ~~acc~~ account for advances in the customer data since an upgrade, and how those might be handled if they are incompatible with the ~~rollback~~ rollback version of warps. ~~for~~.

Using snapshots in this way would allow roll forward and multi-step rollback as well, along with transmission of old versions to Methodics data centers.

Implementation details, cont.

* Factory config reset capability

This is easily implemented for just the warpstor software config data using methods already mentioned.

But perhaps this is about other ~~soft~~ system configurables that get ~~far~~ fucked up. We could use git for this, or even snapshotting of certain directories, or both.

This feature may need to be accessible from grub menu. Which, depending on how it's implemented might ~~req~~ require some early boot time software to be developed to take care of ~~cert~~ certain things.

Q

Does this ~~feature~~ feature include our down ~~load~~ data, yes or no? Do ~~we~~ we wipe all masters @ factory reset time? Two modes: one with and one w/o?

Does this include or not include warpstor config at all?

We could just have a factory
of default tarball that gets un tarred
on a new system snapshot and then
the system reboots. ~~OR~~
OR

We could have a factory default
snapshot that is never actually
run, but just snap slotted, and that snap
is the first version that is run, so
a factory reset could just be an
iteration of that process.