

[RFC v8][PATCH 0/12] Kernel based checkpoint/restart

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-10/msg12191.html>

- *From:* Oren Laadan <orenl@xxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 30 Oct 2008 09:51:03 -0400
-

Basic checkpoint-restart [C/R]: v8 adds support for "external" checkpoint and improves documentation. Older announcements below.

The git tree tracking v8 (branch 'ckpt-v8'), and older versions, is at:
[git://gorgona.ncl.cs.columbia.edu/pub/git/linux-cr-dev.git](http://gorgona.ncl.cs.columbia.edu/pub/git/linux-cr-dev.git)

(or for the latest version –
[git://gorgona.ncl.cs.columbia.edu/pub/git/linux-cr.git](http://gorgona.ncl.cs.columbia.edu/pub/git/linux-cr.git))

We'd like to see these make their way into –mm.
As Dave Hansen put it:

—
Why do we want it? It allows containers to be moved between physical machines' kernels in the same way that VMWare can move VMs between physical machines' hypervisors. There are currently at least two out-of-tree implementations of this in the commercial world (IBM's Metacluster and Parallels' OpenVZ/Virtuozzo) and several in the academic world like Zap.

Why do we need it in mainline now? Because we already have plenty of out-of-tree ones, and want to know what an in-tree one will be like. :) What *I* want right now is the extra review and scrutiny that comes with a mainline submission to make sure we're not going in a direction contrary to the community.

This only supports pretty simple apps. But, I trust Ingo when he says:

Generally, if something works for simple apps already (in a robust, compatible and supportable way) and users find it "very cool", then support for more complex apps is not far in the future. but if you want to support more complex apps straight away, it takes forever and gets ugly.

We're **certainly** going to be changing the ABI (which is the format of the checkpoint). I'd like to follow the model that we used for ext4-dev, which is to make it very clear that this is a development-only feature for now. Perhaps we do that by making the interface only available through debugfs or something similar for now. Or, reserving the syscall numbers but require some runtime switch to be thrown before they can be used. I'm open to suggestions here.

--

Oren.

--

Todo:

- Add support for x86-64 and improve ABI
- Refine or change syscall interface
- Extend to handle (multiple) tasks in a container
- Handle multiple namespaces in a container (e.g. save the filesystem namespaces state with the file descriptors)
- Security (without CAPS_SYS_ADMIN files restore may fail)

Changelog:

[2008-Oct-29] v8:

- Support "external" checkpoint
- Include Dave Hansen's 'deny-checkpoint' patch
- Split docs in Documentation/checkpoint/..., and improve contents

[2008-Oct-17] v7:

- Fix save/restore state of FPU
- Fix argument given to kunmap_atomic() in memory dump/restore

[2008-Oct-07] v6:

- Balance all calls to cr_hbuf_get() with matching cr_hbuf_put() (even though it's not really needed)
- Add assumptions and what's-missing to documentation
- Misc fixes and cleanups

[2008-Sep-11] v5:

- Config is now 'def_bool n' by default
- Improve memory dump/restore code (following Dave Hansen's comments)
- Change dump format (and code) to allow chunks of <vaddr, pages> instead of one long list of each
- Fix use of follow_page() to avoid faulting in non-present pages
- Memory restore now maps user pages explicitly to copy data into them, instead of reading directly to user space; got rid of mprotect_fixup()
- Remove preempt_disable() when restoring debug registers
- Rename headers files s/ckpt/checkpoint/
- Fix misc bugs in files dump/restore
- Fixes and cleanups on some error paths
- Fix misc coding style

[2008-Sep-09] v4:

- Various fixes and clean-ups
- Fix calculation of hash table size
- Fix header structure alignment
- Use stand list_... for cr_pgarr

[2008-Aug-29] v3:

- Various fixes and clean-ups
- Use standard hlist_... for hash table
- Better use of standard kcalloc/kfree

[2008-Aug-20] v2:

- Added Dump and restore of open files (regular and directories)
- Added basic handling of shared objects, and improve handling of 'parent tag' concept
- Added documentation
- Improved ABI, 64bit padding for image data
- Improved locking when saving/restoring memory
- Added UTS information to header (release, version, machine)
- Cleanup extraction of filename from a file pointer
- Refactor to allow easier reviewing
- Remove requirement for CAPS_SYS_ADMIN until we come up with a security policy (this means that file restore may fail)
- Other cleanup and response to comments for v1

[2008-Jul-29] v1:

- Initial version: support a single task with address space of only private anonymous or file-mapped VMAs; syscalls ignore pid/crid argument and act on current process.

At the containers mini-conference before OLS, the consensus among all the stakeholders was that doing checkpoint/restart in the kernel as much as possible was the best approach. With this approach, the kernel will export a relatively opaque 'blob' of data to userspace which can then be handed to the new kernel at restore time.

This is different than what had been proposed before, which was that a userspace application would be responsible for collecting all of this data. We were also planning on adding lots of new, little kernel interfaces for all of the things that needed checkpointing. This unites those into a single, grand interface.

The 'blob' will contain copies of select portions of kernel structures such as vmas and mm_structs. It will also contain copies of the actual memory that the process uses. Any changes in this blob's format between kernel revisions can be handled by an in-userspace conversion program.

This is a similar approach to virtually all of the commercial

[RFC v8][PATCH 0/12] Kernel based checkpoint/restart

checkpoint/restart products out there, as well as the research project Zap.

These patches basically serialize internal kernel state and write it out to a file descriptor. The checkpoint and restore are done with two new system calls: `sys_checkpoint` and `sys_restart`.

In this incarnation, they can only work checkpoint and restore a single task. The task's address space may consist of only private, simple vma's – anonymous or file-mapped. The open files may consist of only simple files and directories.

--

--

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

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

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