Subject: The issues for agreeing on a virtualization/namespaces implementation.
Posted by ebiederm on Tue, 07 Feb 2006 22:06:51 GMT
View Forum Message <> Reply to Message

I think I can boil the discussion down into some of the fundamental
questions that we are facing.

Currently everyone seems to agree that we need something like
my namespace concept that isolates multiple resources.

We need these for
PIDS
UIDS
SYSVIPC
NETWORK
UTSNAME
FILESYSTEM
etc.

The questions seem to break down into:
1) Where do we put the references to the different namespaces?
  - Do we put the references in a struct container that we reference from struct task_struct?
  - Do we put the references directly in struct task_struct?

2) What is the syscall interface to create these namespaces?
  - Do we add clone flags?
    (Plan 9 style)
  - Do we add a syscall (similar to setsid) per namespace?
    (Traditional unix style)?
  - Do we in addition add syscalls to manipulate containers generically?

  I don't think having a single system call to create a container and a new
  instance of each namespace is reasonable as that does not give us a
  path into the future when we create yet another namespace.

  If we have one syscall per each namespace why would we need a container
  structure?

3) How do we refer to namespaces and containers when we are not members?
  - Do we refer to them indirectly by processes or other objects that
    we can see and are members?
  - Do we assign some kind of unique id to the containers?

4) How do we implement each of these namespaces?
  Besides being maintainable are there other constraints?

5) How do we control the resource inside a namespace starting
   from a process that is outside of that namespace?
   - The filesystem mount namespace gave an interesting answer.
     So it is quite possible other namespaces will give
     equally interesting and surprising answers.

6) How do we do all of this efficiently without a noticeable impact on
   performance?
   - I have already heard concerns that I might be introducing cache
     line bounces and thus increasing tasklist_lock hold time.
     Which on big way systems can be a problem.

7) How do we allow a process inside a container to create containers
   for it's children?
   - In general this is trivial but there are a few ugly issues
     here.

I think these are the key questions of the conversation.

Personally so long as we get true namespaces, implemented in a
performant and maintainable way that a process from the inside can't
distinguish from what we have now I have no hard requirements.

Eric