- **Regan Robertson** Mod • 14 days ago
  The video will start on this page at 9:00am, and you may have to hit play or unmute. As a reminder this site works best in a Chrome Browser. You can write in comments through this feature to continue the conversation or ask questions. Please login or sign up for a Disqus account to participate in the discussion boards.
  -
  -
  - Reply
  -
  - Share ›
  -
-
  -
  -
    
    Hui Lu • 14 days ago
    Good morning! Thanks for checking out this talk. I'd like to take any questions and comments during this session.
    - 1
    -
    - Reply
    -
    - Share ›
    -
      -
        -
        -
          
          Renato Levy Hui Lu • 14 days ago
          Good morning! Wow, I can't believe this is already the last day of the Summit.
          -
          -
          - Reply
          -
          - Share ›
          -
-
  -
  -

Jason H Li • 14 days ago

We talked about microservice and containerization issues during Day 1. Question - architecting wise, what's the benefit of putting tiny VM on top of KVM, which itself is a Type 2 hypervisor that needs a host? This stacking is a bit of confusing to me.

- Reply
- Share ›

Hui Lu Jason H Li • 14 days ago

Jason, this is a good question. Yes, it is not necessary to put tiny VM upon KVM. Rather, it would be better to eliminate KVM from the design. We are working on this. However, the current one is only for preliminary testing purpose.

- Reply
- Share ›

Jason H Li Hui Lu • 14 days ago

If only for prelim testing, that's fine, as a convenience for setup.

- 1
- Reply
- Share ›

Hui Lu Jason H Li • 14 days ago

Our goal is to run seL4 secured micro-services without additional virtual hardware abstraction (i.e., no KVM and/or QEMU) thus eliminating those unnecessary stacks. This is similar to the method introduced in this paper that runs unikernels as processes: https://dl.acm.org/doi/10.1....

Reply

Share ›

Jason H Li Hui Lu • 14 days ago

Yep that sounds good.

Reply

Share ›

Ihor Kuz • 14 days ago

It sounds like you are proposing running seL4 on QEMU on an existing hypervisor. This means that the QEMU + hypervisor + hardware are part of the TCB. This is huge and removes any benefit that a (verified) seL4 has. What would be the point of using seL4 in this way?

Reply

Share ›

Jason H Li Ihor Kuz • 14 days ago

Ihor - I think we had similar concerns here - see my question below.

Reply

Share ›

Ihor Kuz Jason H Li • 14 days ago

Yes, I saw yours after I submitted.

Reply

Share ›

Ihor Kuz Ihor Kuz • 14 days ago

If you replaced the "OS/hypervisor" with seL4, and removed seL4 from your tiny VM instances, running "rumprun" directly on seL4, then you would significantly reduce the TCB.

- Reply
- Share ›

Renato Levy Ihor Kuz • 14 days ago

my point exactly

- Reply
- Share ›

Jason H Li Ihor Kuz • 14 days ago

I think I have the same pic in my mind.

- Reply
- Share ›

Hui Lu Ihor Kuz • 14 days ago

Ihor, thanks for the question. This is limited by the current cloud infrastructure which uses commodity machines plus QEMU and hypervisor to provide virtualized running environments (VMs) to end users. We are investigating the possibility of using seL4 in such an existing setup instead of re-designing the whole infrastructure. As the first step, we would like to provide a stronger isolation virtual running environment with seL4 -- the tiny seL4 VM. We'd like to compare it with existing other isolation techniques like VMs, containers, AWS firecracker, etc.

Reply

Share ›

Ihor Kuz Hui Lu • 14 days ago • edited

but running seL4 in a VM doesn't improve isolation in any way (compared to the isolation that the VM provides already). It seems like using seL4 in this way just makes things harder, adds more overhead, but does not add any isolation benefit.

Reply

Share ›

Hui Lu Ihor Kuz • 14 days ago • edited

Currently in cloud platforms, we have two popular isolation methods: VMs vs. Containers. For micro-services platforms, it is desired to run lightweight containers for the performance purpose. However, its isolation is weaker than VMs. So people are looking for some "in-between" solution -- as lightweight as containers while as secure as VMs. The current way for such tiny VMs are to minimalize VM kernel. We think we might replace the traditional monolithic kernel with seL4 kernel. We are working on how to further mitigate (or eliminate) the overhead of KVM and QEMU.

Reply

Share ›

Ihor Kuz Hui Lu • 14 days ago • edited

Yes, that (minimising the hypervisor) would make sense. But that's not what your proposal does. My understanding of what you've presented is that it adds seL4 on top of the 'heavyweight' VM, but fundamentally relies on the VM for isolation.

Reply

Share ›

Hui Lu Ihor Kuz • 14 days ago

Sorry for the confusion. Adding seL4 on top of the 'heavyweight' VM in the current design is for preliminary testing purpose only (see my answer to Jason below). In our proposed work, we mentioned to eliminate the 'heavyweight' VM with much lightweight, specialized QEMU and further allow one QEMU to manage/control multiple seL4 unikernels.

- Reply
- Share ›

Ihor Kuz Hui Lu • 14 days ago

But you still rely on QEMU, and you're still not using any seL4 features (if you run multiple seL4 instances, with a single rumprun per seL4), so it doesn't really change my concerns. What would change if you removed seL4 from your target design?

- Reply
- Share ›

Hui Lu Ihor Kuz • 14 days ago

Yes, that concern makes sense to me and it would be helpful for us to think more about how seL4 features can be better leveraged. We do need QEMU thus far as we target running multiple seL4 unikernel instances on the same physical machine. In this sense, we use seL4 because it is one of microkernels with high-quality code (no bugs?). In this case, we are investigating the seL4 microkernel option for tiny VMs. As I replied to Jason below, we investigate how to further eliminate QEMU and KVM to run seL4 unikernels as "processes" (https://dl.acm.org/doi/10.1.... In this case, there are no VMs and seL4 layer could separate the application functions from the underlying OSes (similar to google gvisor -- a regulation layer sitting between user applications and OSes). But I believe it needs to modify the code of seL4 (I am not sure how feasible it is in the seL4 case where formal proof plays a key role).

Reply

Share ›

Jason H Li • 14 days ago

If you containerize apps and make a good package for seL4, I can see good uses for various apps running more efficiently directly on seL4 (on hardware) than using a VMM as we do nowadays. At least that's how I see it at this moment. Not on top of Type 2 HV - host - hardware.

Reply

Share ›

This comment was deleted.

Jason H Li Guest • 14 days ago

Possible, I kind of figure that, but was still puzzled, architecting wise.

Reply

Share ›

Ihor Kuz Guest • 14 days ago

It could be, but then why use seL4?

Reply

Share ›

This comment was deleted.

Ihor Kuz Guest • 14 days ago

In this case you could run rumprun or a unikernel directly. You are not using any of the features of seL4.

Reply

Share ›

Hui Lu Guest • 14 days ago • edited

Lok. Thanks for making my point clear here. Yes, we are investigating an seL4-enabled sandbox technique on existing cloud FaaS infrastructure.

Reply

Share ›

Renato Levy • 14 days ago

Why not build so that the tiny image can run natively on top of seL4 as the hypervisor (it would require just a small adapter). It would be much faster to boot, and still assure separation.

Reply

Share ›

Renato Levy Renato Levy • 14 days ago

This is what i was talking about yesterday on executing a JVM on top of seL4

Reply

Share ›

Hui Lu Renato Levy • 14 days ago

Thanks Renato. This is a good point and should be the desired design in terms of security, performance, and simplicity. I guess the difficulty here is that the cloud has mature ecosystem which closely relies on traditional OSes (Linux/windows/hypervisors). In addition to running isolated applications/functions, we do need other cloud-related services running on the same machine. So instead of having a clean-stale solution, we are investigating how seL4 can contribute to an existing cloud ecosystem. And we are now focused on applying seL4 to build a (relatively) lightweight tiny VM for executing in-cloud functions.

Reply

Share ›

Regan Robertson Mod • 14 days ago

Please join us for the next session that starts in 2 minutes. You can either go back to agenda to get to the next session or at the bottom of the page there is a next session button.

- Reply
- Share ›