Hello! I’m Michael Tsirkin. I work at Red Hat as a distinguished engineer and I’m a chair of the Virtio Technical committee.

Today I’m going to talk about the work we are doing in VirtIO for the benefit of hardware VirtIO implementations.

I think ViirtIO is kind of unusual in that it has been defined as a software interface first of all, and hardware came afterwards, so we have hardware emulating software in a sense. I plan to describe some challenges that surface when you try to do it like this. Hopefully this is going to be interesting for people interested in both software and hardware.
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“ Naturally, the hardware would be built to provide the features that existing guests use, since that’s the software people want to speed up. In response, we on the Virtio side of things, noticed this trend and we thought: okay that’s great, isn’t there anything we can do to make hardware work better? We’ll put it in the spec, we’ll build it into software, and then in a couple of years after the software is widely deployed it will be worth it for hardware to target these interface extensions. These extensions is what this presentation is about.
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“ Naturally, the hardware would be built to provide the features that existing guests use, since that’s the software people want to speed up.

In response, we on the Virtio side of things, noticed this trend and we thought: okay that’s great, isn’t there anything we can do to make hardware work better?

We’ll put it in the spec, we’ll build it into software, and then in a couple of years after the software is widely deployed it will be worth it for hardware to target these interface extensions. These extensions is what this presentation is about.
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

Naturally, the hardware would be built to provide the features that existing guests use, since that’s the software people want to speed up.

In response, we on the Virtio side of things, noticed this trend and we thought: okay that’s great, isn’t there anything we can do to make hardware work better?

We’ll put it in the spec, we’ll build it into software, and then in a couple of years after the software is widely deployed it will be worth it for hardware to target these interface extensions.

These extensions is what this presentation is about.
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

Naturally, the hardware would be built to provide the features that existing guests use, since that’s the software people want to speed up.

In response, we on the Virtio side of things, noticed this trend and we thought: okay that’s great, isn’t there anything we can do to make hardware work better?

We’ll put it in the spec, we’ll build it into software, and then in a couple of years after the software is widely deployed it will be worth it for hardware to target these interface extensions.

These extensions is what this presentation is about.
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“
Naturally, the hardware would be built to provide the features that existing guests use, since that’s the software people want to speed up.
In response, we on the Virtio side of things, noticed this trend and we thought: okay that’s great, isn’t there anything we can do to make hardware work better?
We’ll put it in the spec, we’ll build it into software, and then in a couple of years after the software is widely deployed it will be worth it for hardware to target these interface extensions.
These extensions is what this presentation is about.
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“ Naturally, the hardware would be built to provide the features that existing guests use, since that’s the software people want to speed up.

In response, we on the Virtio side of things, noticed this trend and we thought: okay that’s great, isn’t there anything we can do to make hardware work better?

We’ll put it in the spec, we’ll build it into software, and then in a couple of years after the software is widely deployed it will be worth it for hardware to target these interface extensions. These extensions is what this presentation is about.
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?”

Naturally, the hardware would be built to provide the features that existing guests use, since that’s the software people want to speed up.

In response, we on the Virtio side of things, noticed this trend and we thought: okay that’s great, isn’t there anything we can do to make hardware work better?

We’ll put it in the spec, we’ll build it into software, and then in a couple of years after the software is widely deployed it will be worth it for hardware to target these interface extensions.

These extensions is what this presentation is about.
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“
Naturally, the hardware would be built to provide the features that existing guests use, since that’s the software people want to speed up.
In response, we on the Virtio side of things, noticed this trend and we thought: okay that’s great, isn’t there anything we can do to make hardware work better?
We’ll put it in the spec, we’ll build it into software, and then in a couple of years after the software is widely deployed it will be worth it for hardware to target these interface extensions.
These extensions is what this presentation is about.
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“
Naturally, the hardware would be built to provide the features that existing guests use, since that’s the software people want to speed up.
In response, we on the Virtio side of things, noticed this trend and we thought: okay that’s great, isn’t there anything we can do to make hardware work better?
We’ll put it in the spec, we’ll build it into software, and then in a couple of years after the software is widely deployed it will be worth it for hardware to target these interface extensions.
These extensions is what this presentation is about.
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“ Naturally, the hardware would be built to provide the features that existing guests use, since that’s the software people want to speed up. In response, we on the Virtio side of things, noticed this trend and we thought: okay that’s great, isn’t there anything we can do to make hardware work better? We’ll put it in the spec, we’ll build it into software, and then in a couple of years after the software is widely deployed it will be worth it for hardware to target these interface extensions. These extensions is what this presentation is about.
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“ Naturally, the hardware would be built to provide the features that existing guests use, since that’s the software people want to speed up. In response, we on the Virtio side of things, noticed this trend and we thought: okay that’s great, isn’t there anything we can do to make hardware work better? We’ll put it in the spec, we’ll build it into software, and then in a couple of years after the software is widely deployed it will be worth it for hardware to target these interface extensions. These extensions is what this presentation is about.
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

Naturally, the hardware would be built to provide the features that existing guests use, since that’s the software people want to speed up.

In response, we on the Virtio side of things, noticed this trend and we thought: okay that’s great, isn’t there anything we can do to make hardware work better?

We’ll put it in the spec, we’ll build it into software, and then in a couple of years after the software is widely deployed it will be worth it for hardware to target these interface extensions.

These extensions is what this presentation is about.
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

Naturally, the hardware would be built to provide the features that existing guests use, since that’s the software people want to speed up.

In response, we on the Virtio side of things, noticed this trend and we thought: okay that’s great, isn’t there anything we can do to make hardware work better?

We’ll put it in the spec, we’ll build it into software, and then in a couple of years after the software is widely deployed it will be worth it for hardware to target these interface extensions.

These extensions is what this presentation is about.
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“ Naturally, the hardware would be built to provide the features that existing guests use, since that’s the software people want to speed up. In response, we on the Virtio side of things, noticed this trend and we thought: okay that’s great, isn’t there anything we can do to make hardware work better? We’ll put it in the spec, we’ll build it into software, and then in a couple of years after the software is widely deployed it will be worth it for hardware to target these interface extensions. These extensions is what this presentation is about.
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“
Naturally, the hardware would be built to provide the features that existing guests use, since that’s the software people want to speed up.
In response, we on the Virtio side of things, noticed this trend and we thought: okay that’s great, isn’t there anything we can do to make hardware work better?
We’ll put it in the spec, we’ll build it into software, and then in a couple of years after the software is widely deployed it will be worth it for hardware to target these interface extensions.
These extensions is what this presentation is about.
To summarize, a large existing ecosystem makes Virtio a compelling option for new devices. We are looking to enhance the way we handle hardware Virtio devices, with work on both performance and new features. Contribution is open to everyone – so join us!
To summarize, a large existing ecosystem makes Virtio a compelling option for new devices. We looking to enhance the way we handle hardware Virtio devices, with work on both performance and new features. Contribution is open to everyone – so join us!
This ends the talk, if you have any questions feel free to reach out. The best way to do this is to post them on the virtio mailing list since it is preferable to have all discussion happen in the open. Thank you for your patience, and have a good day!