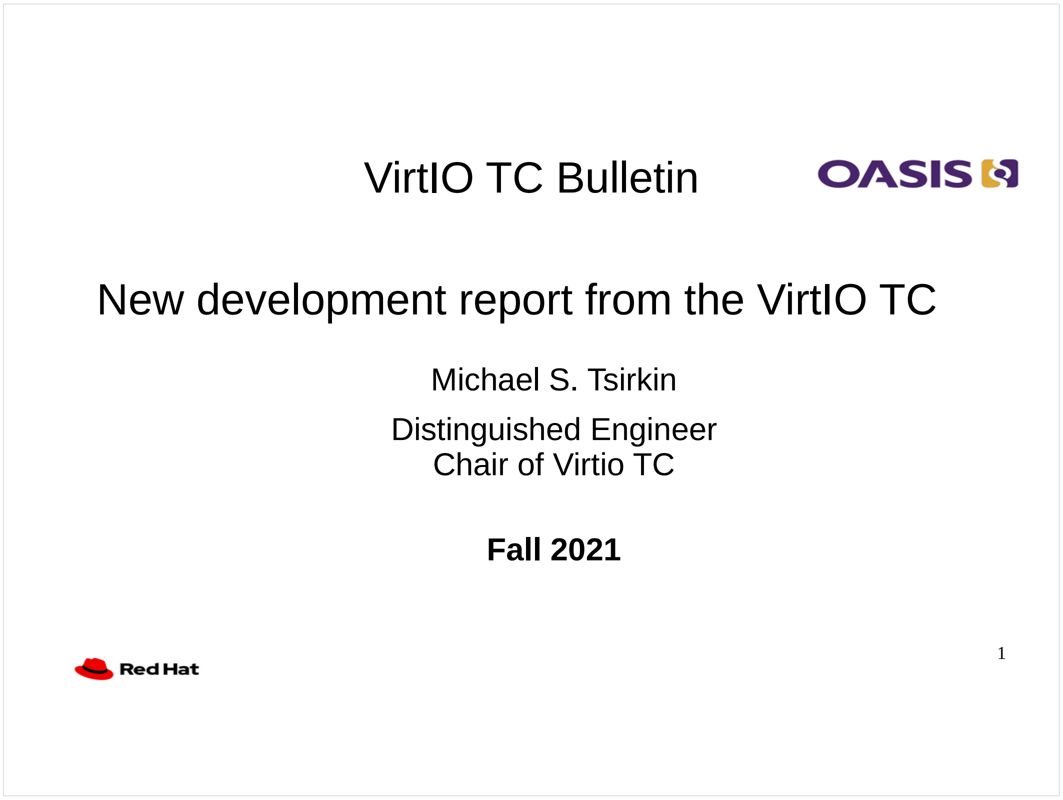


VirtIO TC Bulletin

New development report from the VirtIO TC

Michael S. Tsirkin

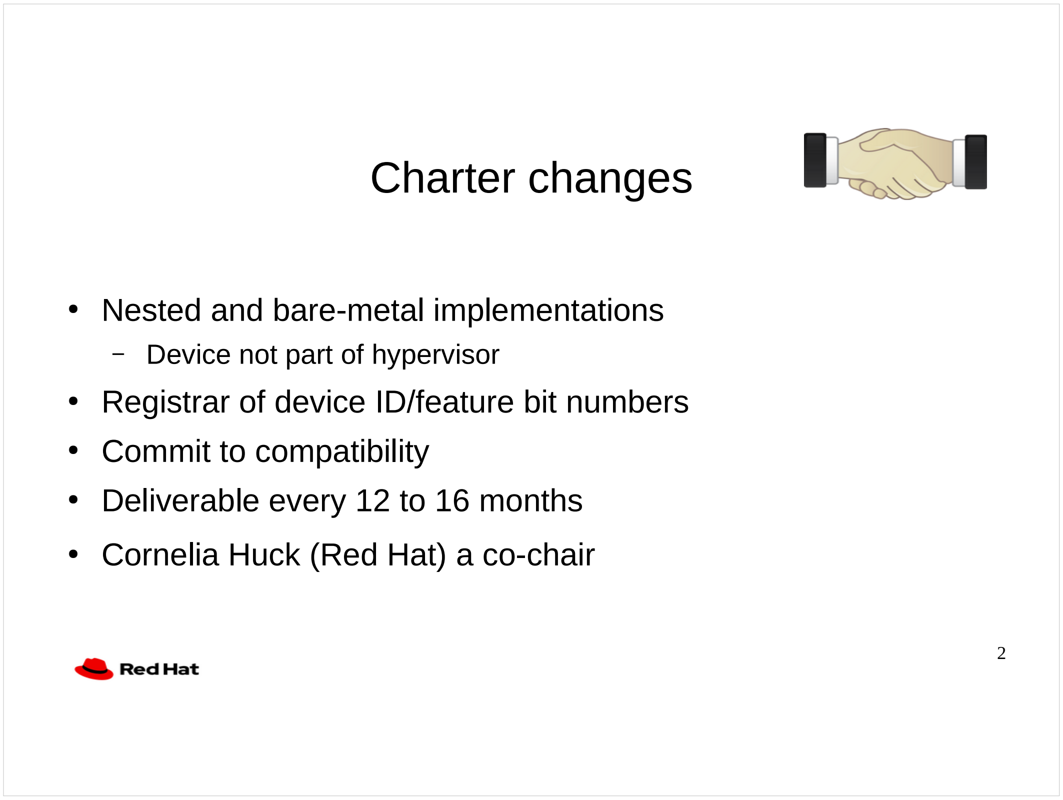
Distinguished Engineer
Chair of Virtio TC

Fall 2021

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.



Charter changes

- Nested and bare-metal implementations
  - Device not part of hypervisor
- Registrar of device ID/feature bit numbers
- Commit to compatibility
- Deliverable every 12 to 16 months
- Cornelia Huck (Red Hat) a co-chair

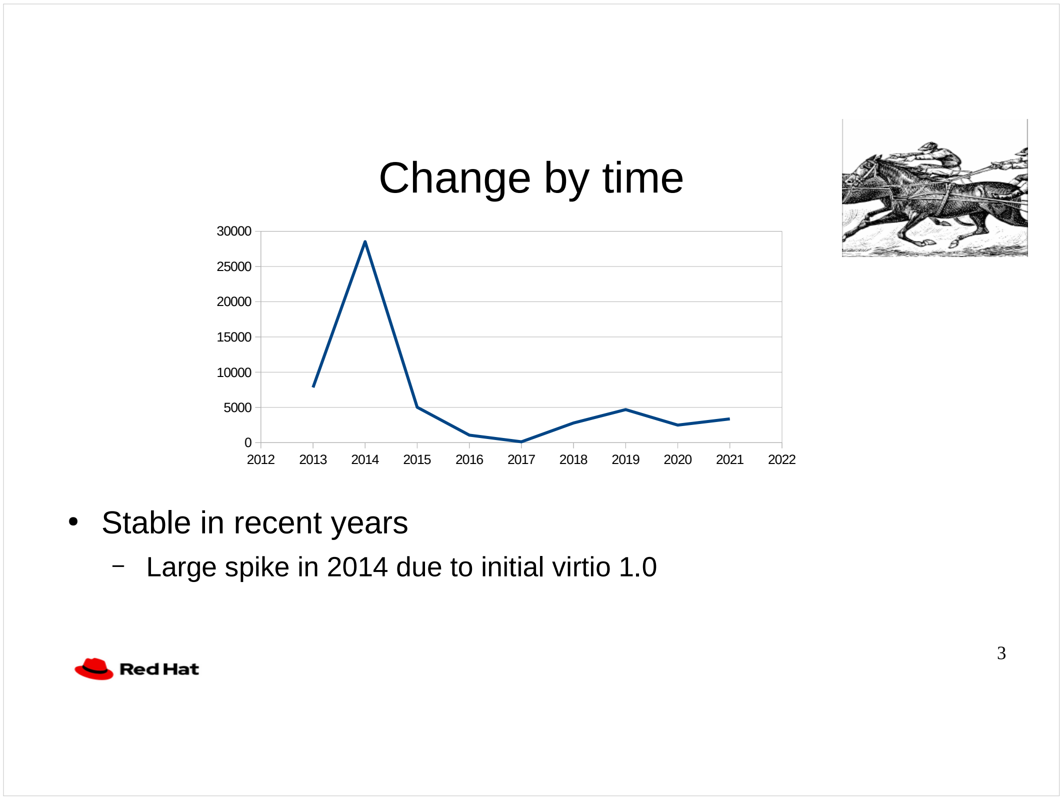
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

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.



Change by time

- Stable in recent years
  - Large spike in 2014 due to initial virtio 1.0

3

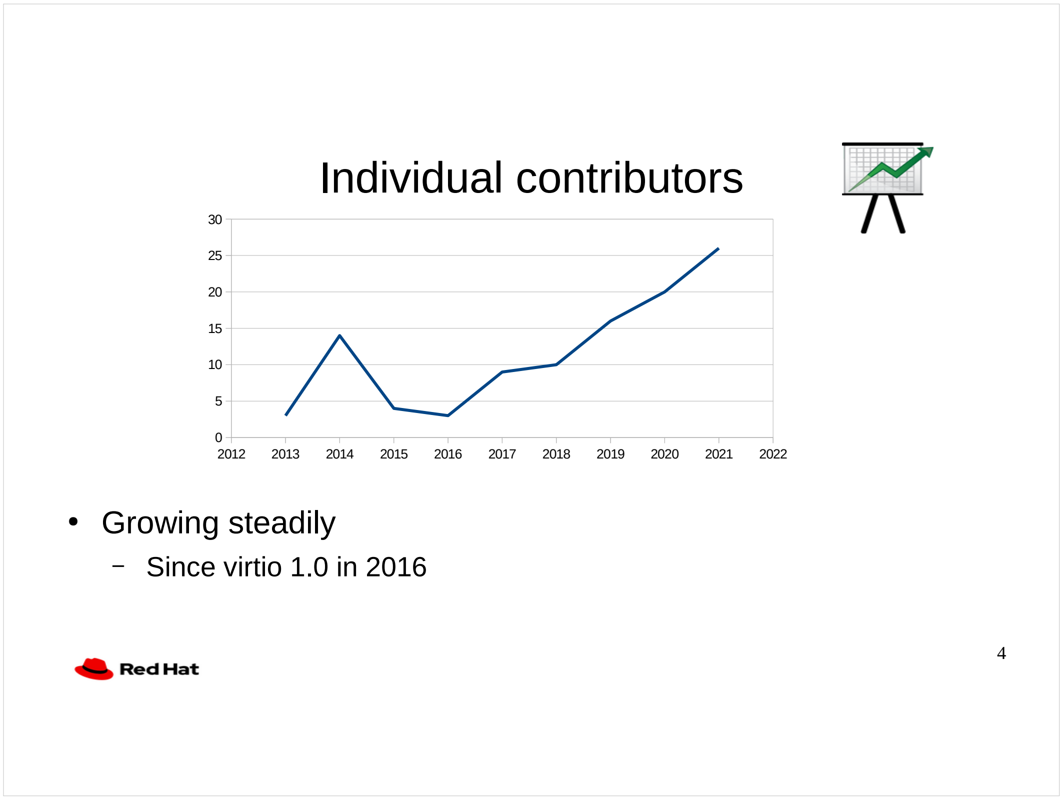
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

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.



Individual contributors

- Growing steadily
  - Since virtio 1.0 in 2016

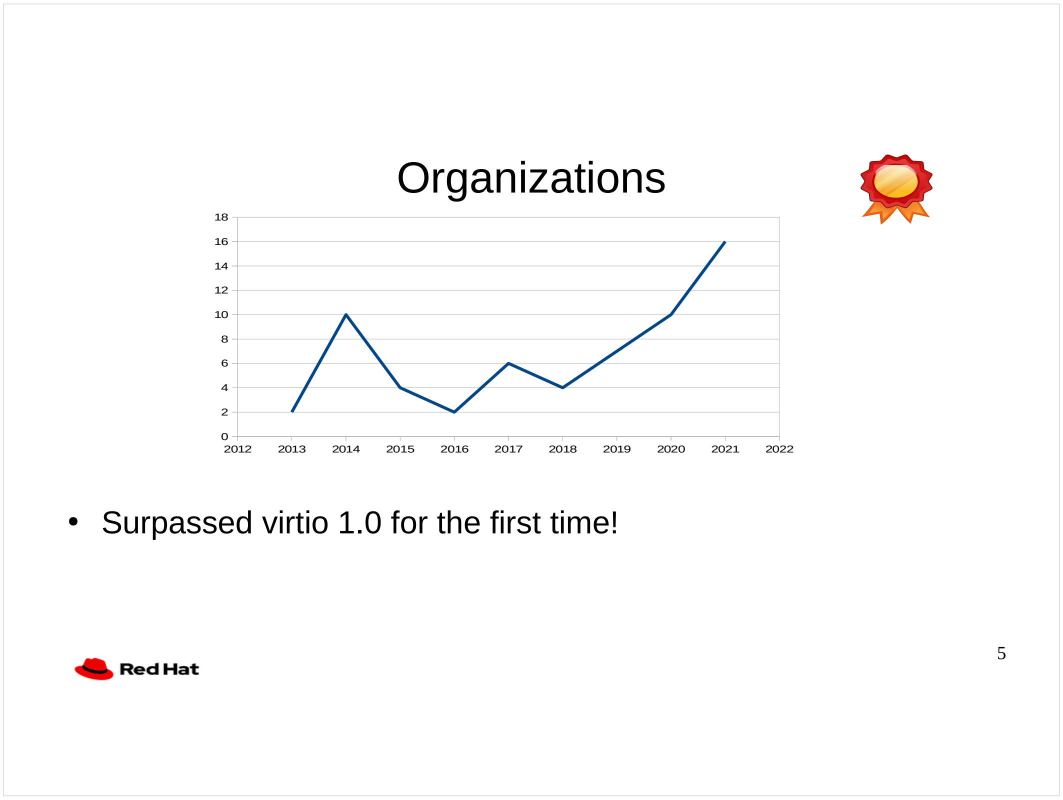
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

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.



- Surpassed virtio 1.0 for the first time!

5

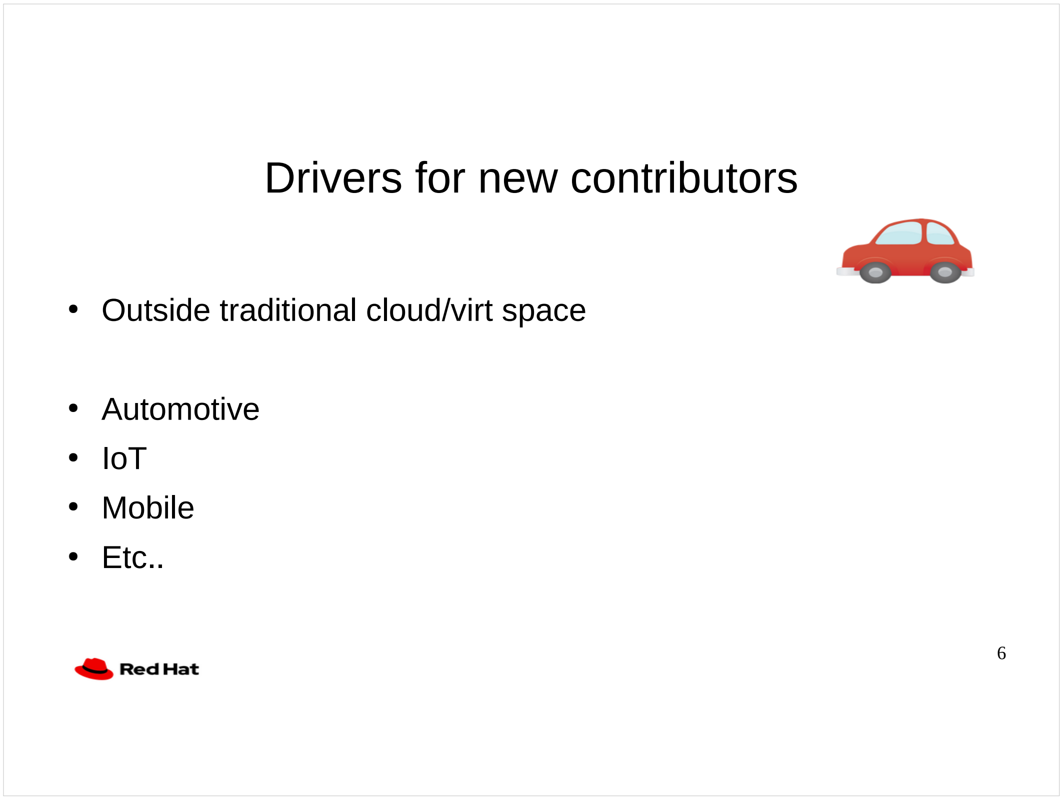
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

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.



Drivers for new contributors

- Outside traditional cloud/virt space

- Automotive
- IoT
- Mobile
- Etc..

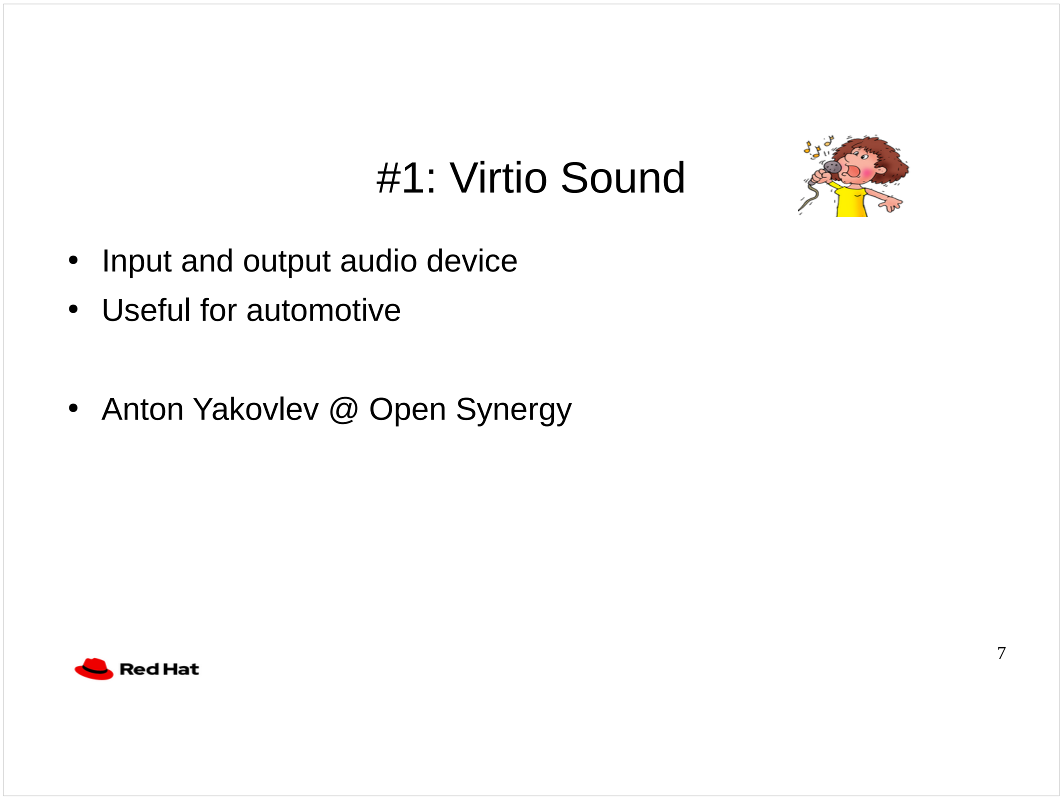
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

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.



#1: Virtio Sound

- Input and output audio device
- Useful for automotive

- Anton Yakovlev @ Open Synergy

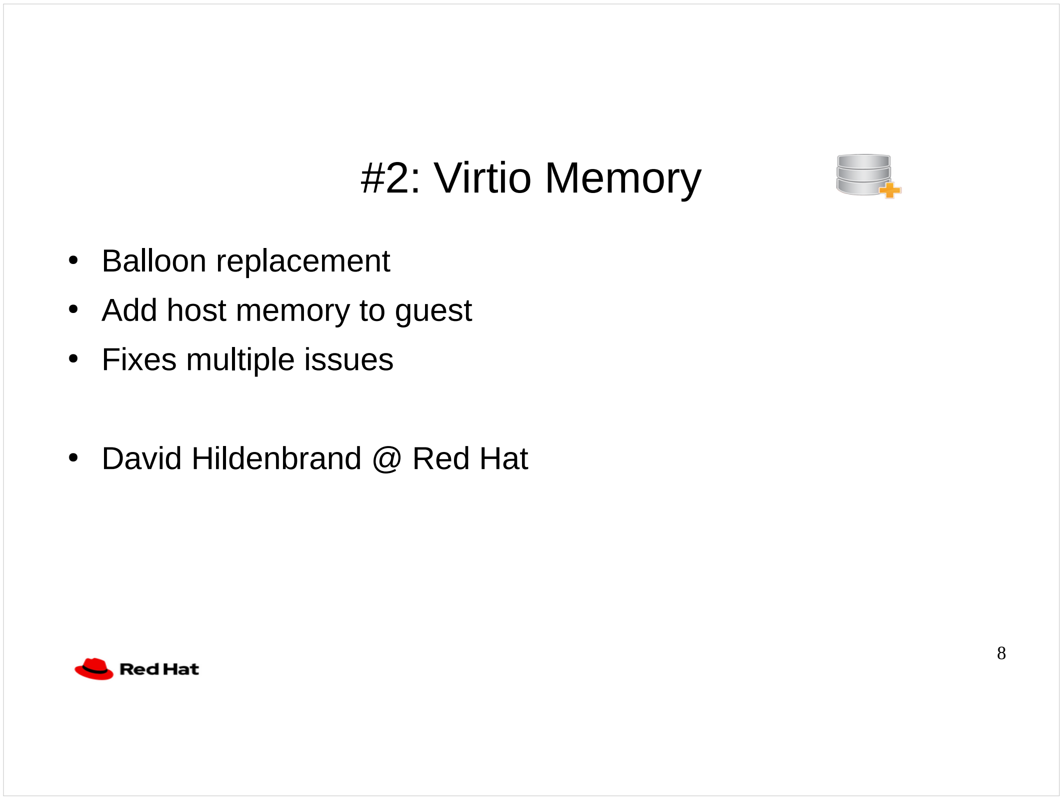
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

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.



#2: Virtio Memory

- Balloon replacement
- Add host memory to guest
- Fixes multiple issues

- David Hildenbrand @ Red Hat

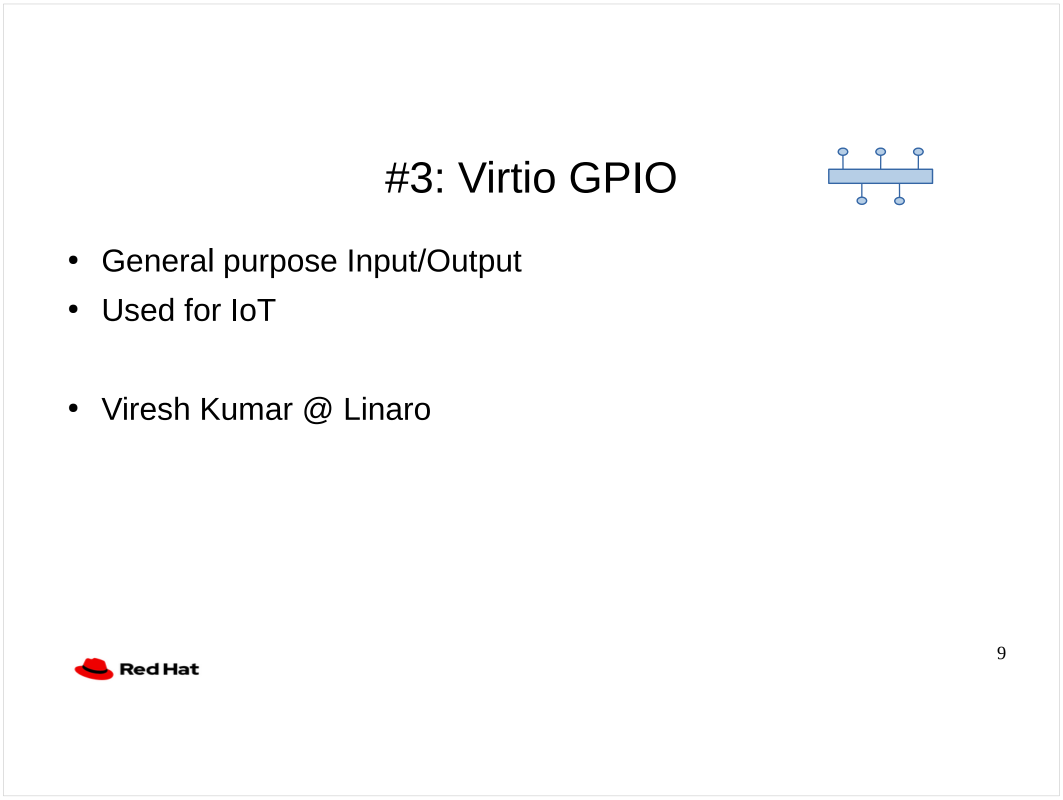
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

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.



#3: Virtio GPIO

- General purpose Input/Output
- Used for IoT

- Viresh Kumar @ Linaro

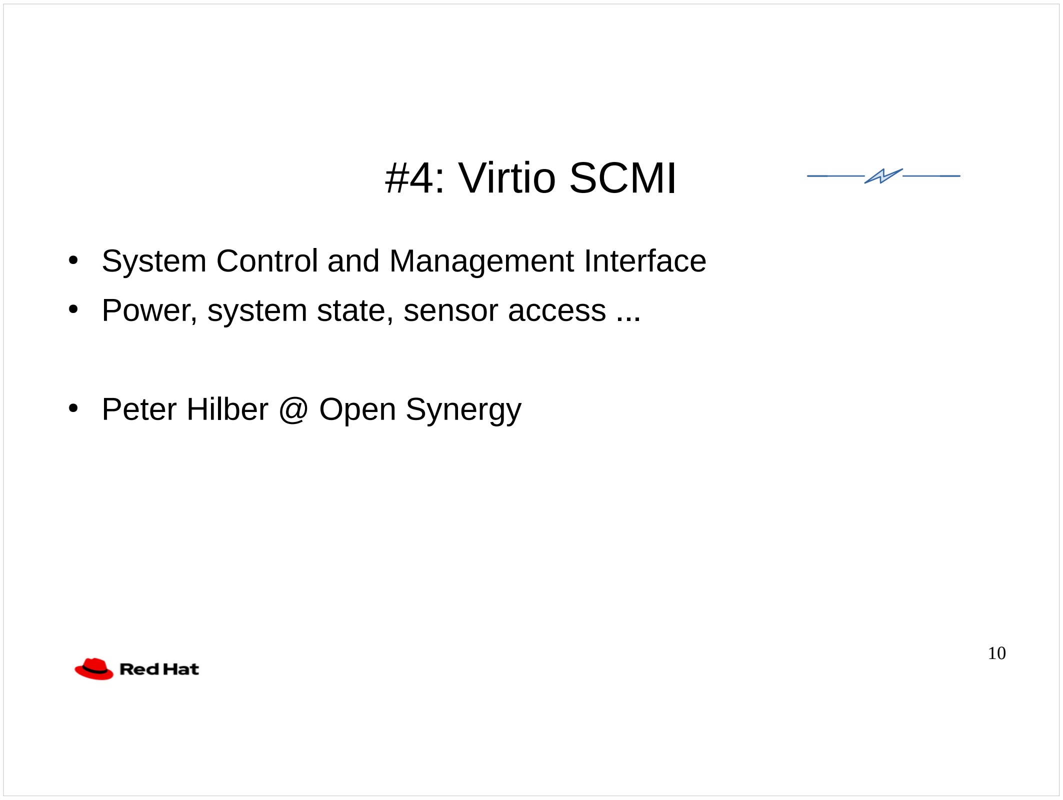
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

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.



#4: Virtio SCMI

- System Control and Management Interface
- Power, system state, sensor access ...

- Peter Hilber @ Open Synergy

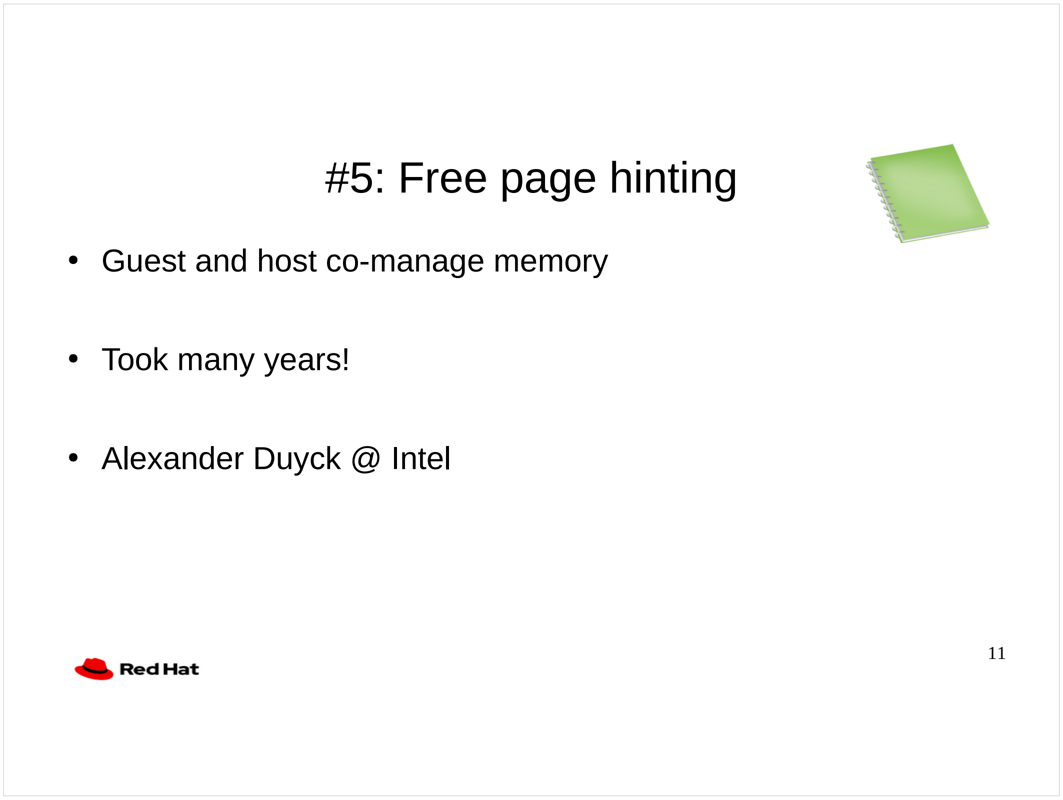
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

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.



#5: Free page hinting

- Guest and host co-manage memory
- Took many years!
- Alexander Duyck @ Intel

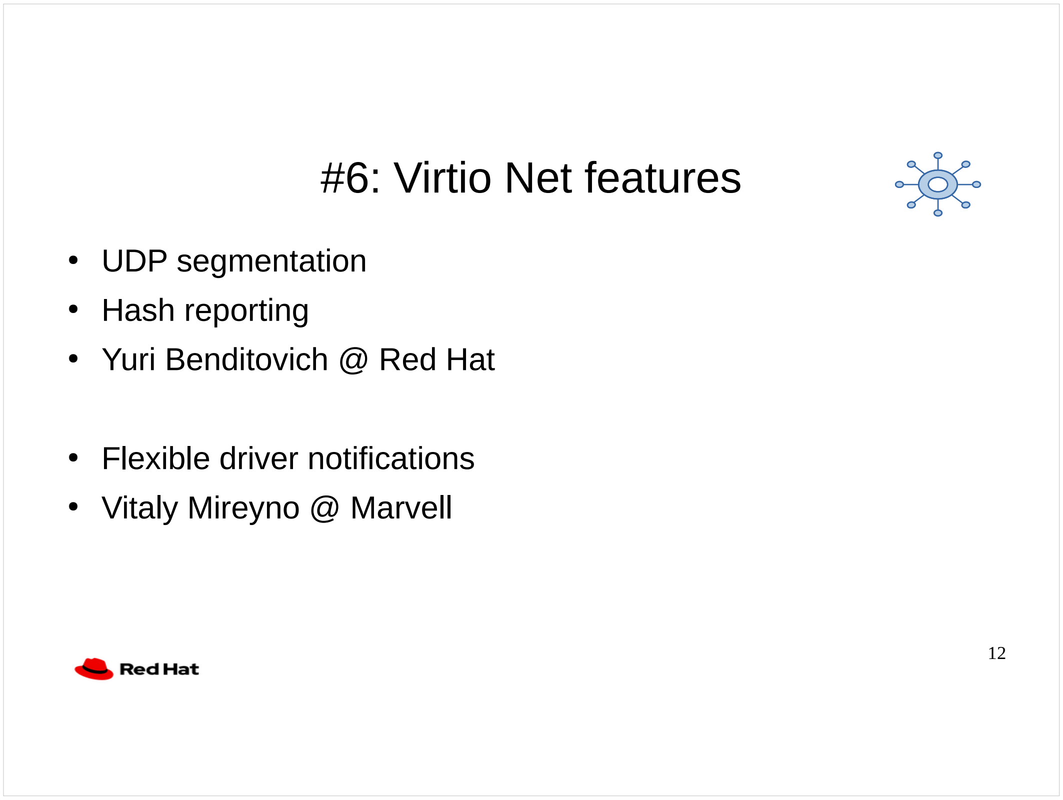
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

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.



#6: Virtio Net features

- UDP segmentation
- Hash reporting
- Yuri Benditovich @ Red Hat

- Flexible driver notifications
- Vitaly Mireyno @ Marvell

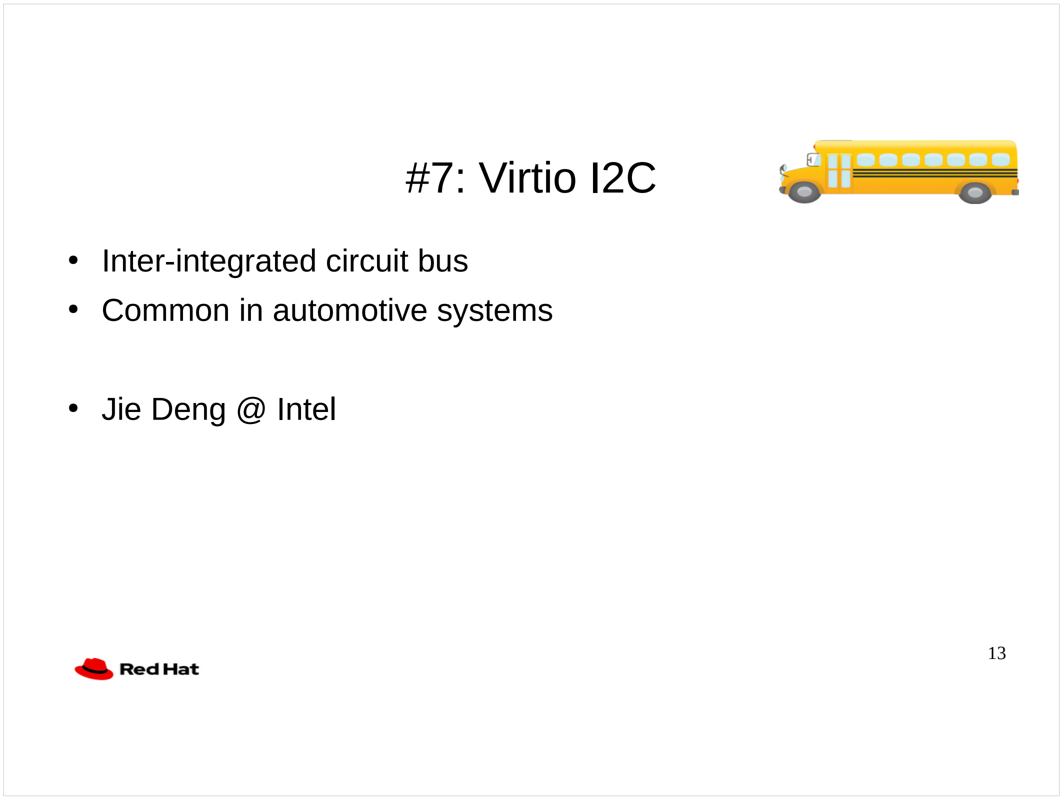
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

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.



#7: Virtio I2C

- Inter-integrated circuit bus
- Common in automotive systems

- Jie Deng @ Intel

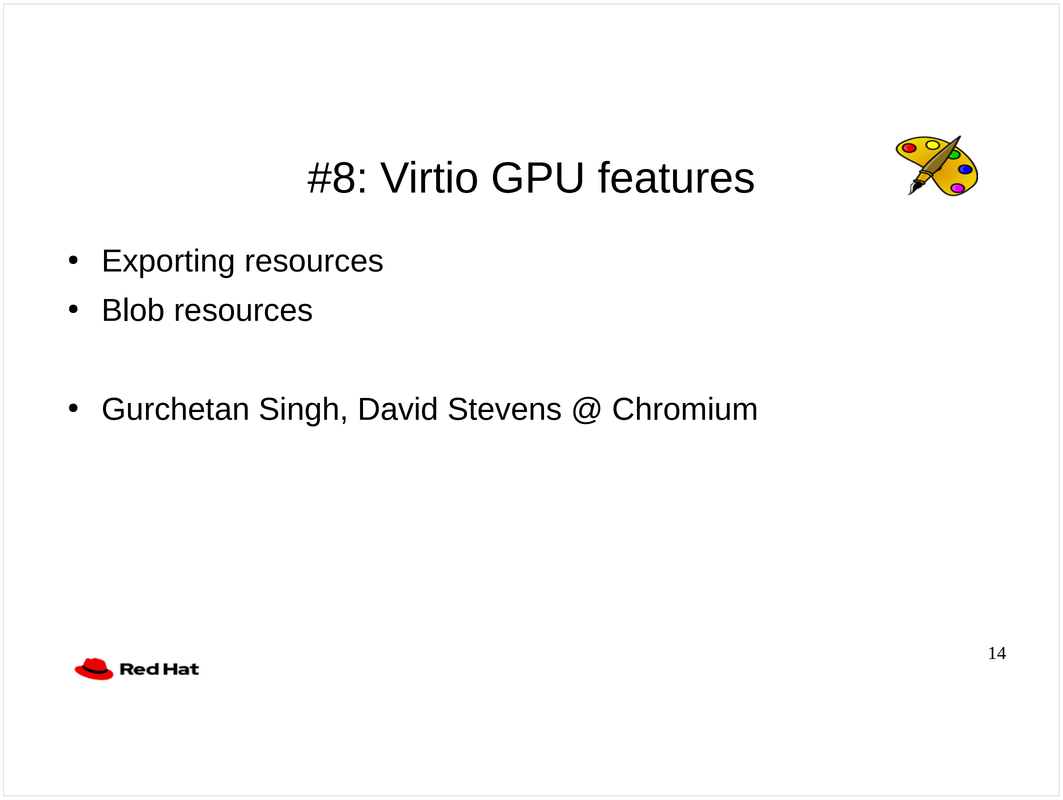
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

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.



#8: Virtio GPU features

- Exporting resources
- Blob resources

- Gurchetan Singh, David Stevens @ Chromium

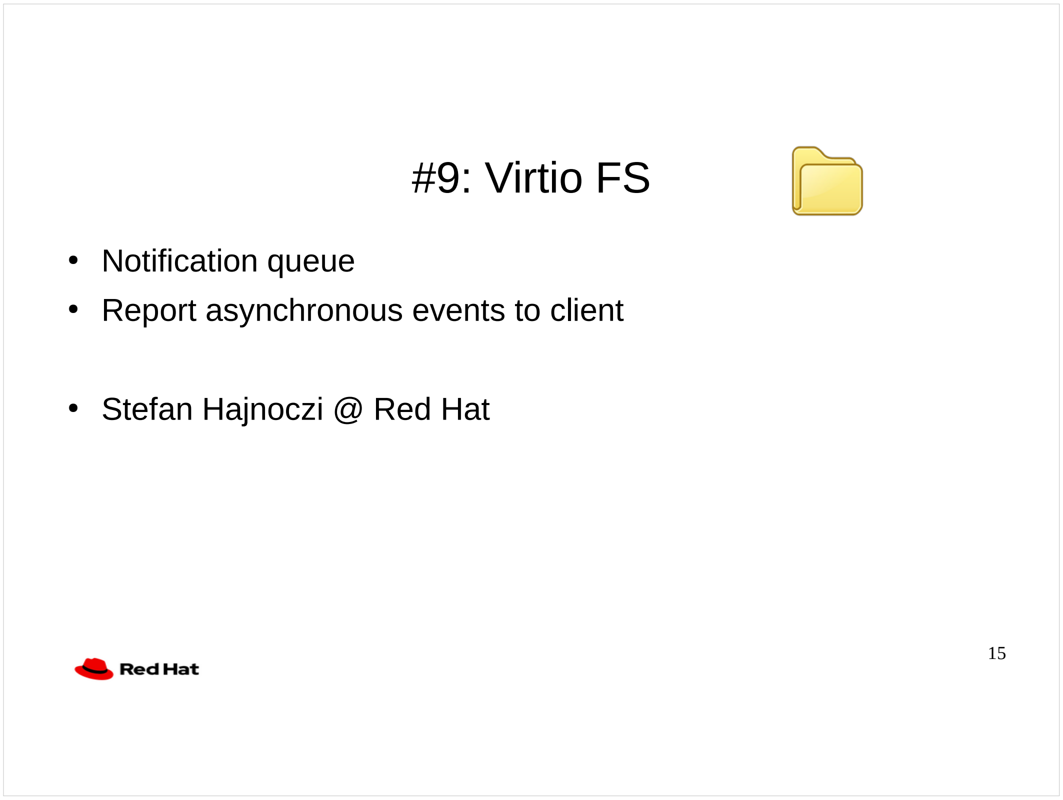
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

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.



#9: Virtio FS

- Notification queue
- Report asynchronous events to client

- Stefan Hajnoczi @ Red Hat

Red Hat

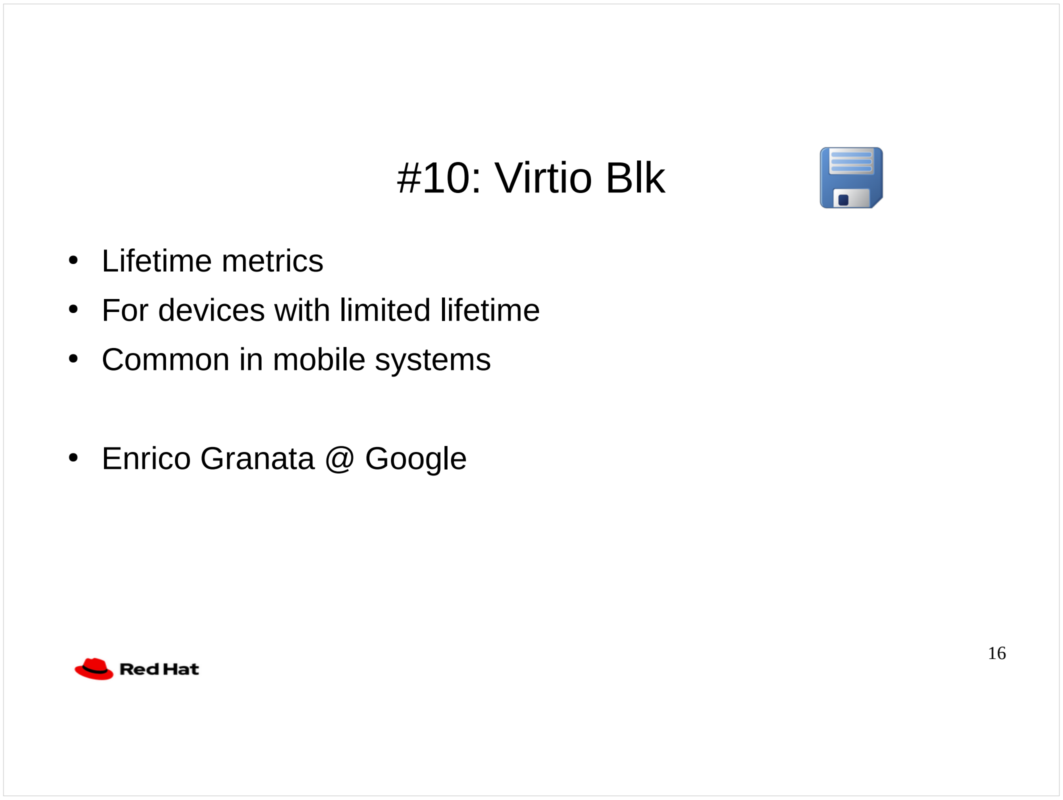
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

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.



#10: Virtio Blk

- Lifetime metrics
- For devices with limited lifetime
- Common in mobile systems

- Enrico Granata @ Google

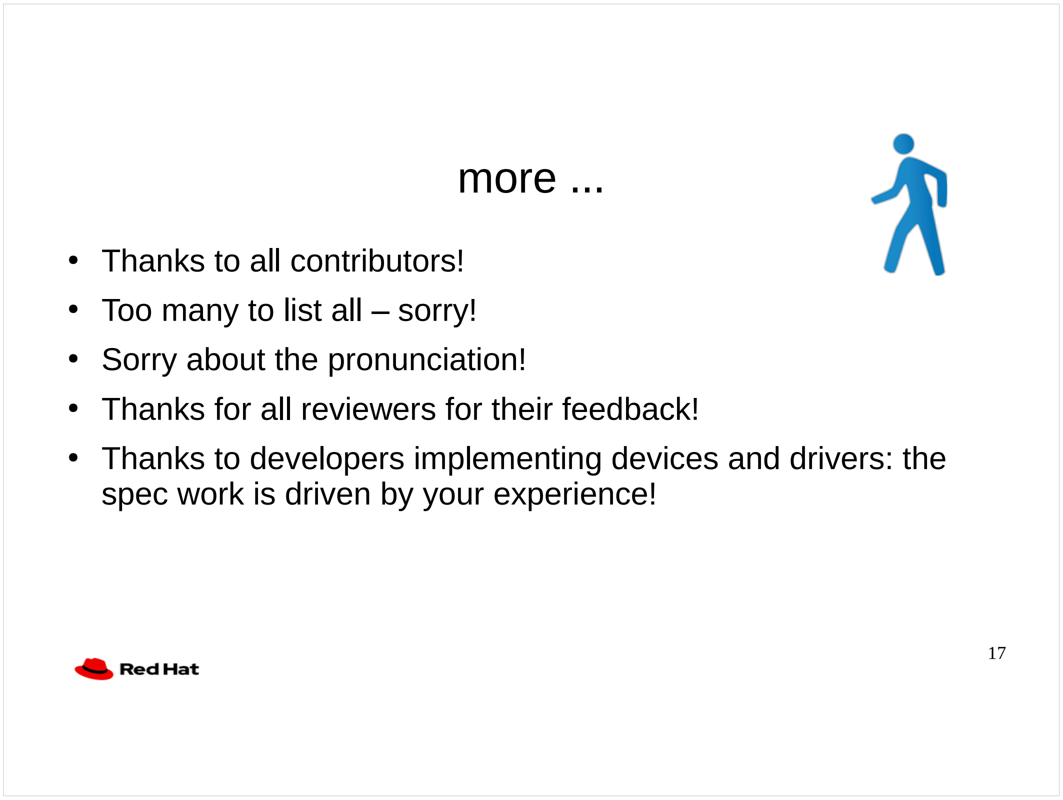
So it all began with a quest for performance, where a cloud vendor would say “right now I’m burning up host CPU cycles moving packets between a virtio device and a host NIC, how about we teach the NIC to talk virtio directly instead?“

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.



more ...

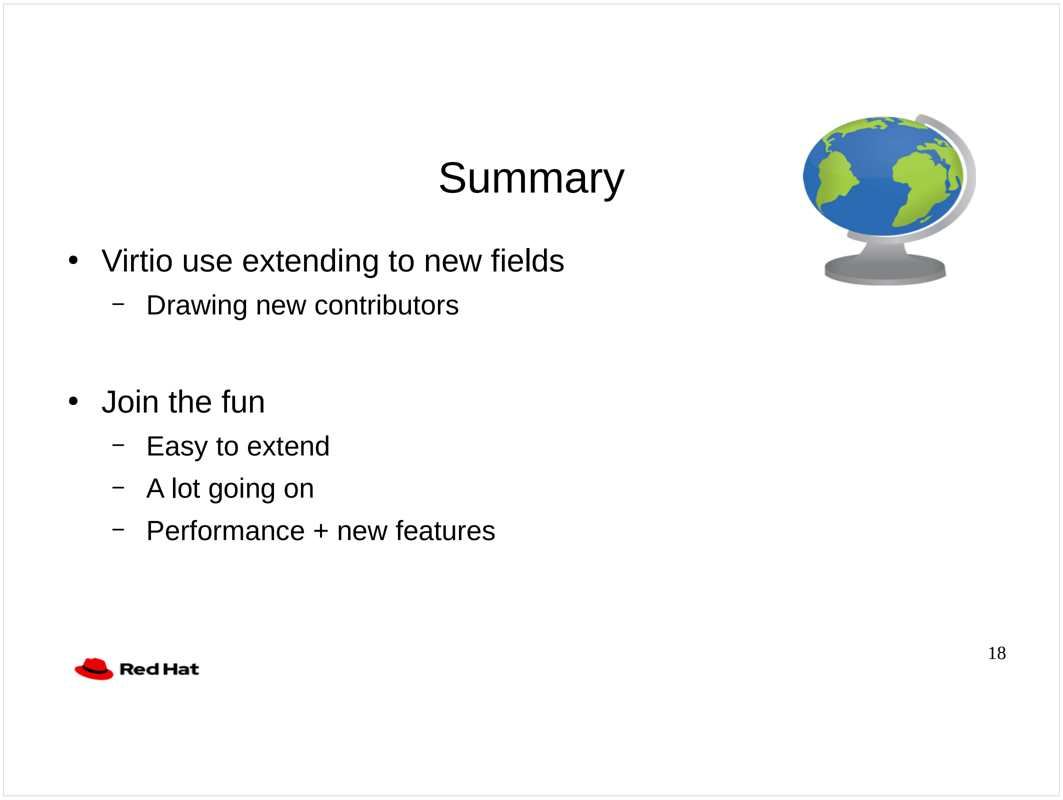
- Thanks to all contributors!
- Too many to list all – sorry!
- Sorry about the pronunciation!
- Thanks for all reviewers for their feedback!
- Thanks to developers implementing devices and drivers: the spec work is driven by your experience!

Red Hat

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 workon both performance and new features.

Contribution is open to everyone – so join us!



Summary

- Virtio use extending to new fields
  - Drawing new contributors
- Join the fun
  - Easy to extend
  - A lot going on
  - Performance + new features

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 workon both performance and new features.

Contribution is open to everyone – so join us!

Questions?

- Subscribe to virtio-dev@lists.oasis.org
- Send questions to virtio-dev@lists.oasis.org
  - Copy relevant people

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!