How many threads does it take for Zephyr to choke? Paraphrased, what are the performance limitations that are known?
regarding 'choking' on threads, zephyr's scheduling algorithm is very configurable and scales up to large numbers of threads. you can read details here:
https://docs.zephyrproject.org/latest/reference/kernel/scheduling/index.html
sorry, bad copy/paste there -- the scheduling algorithm is very configurable and scales to a large number of threads is what I meant to say

Will OpenThread and CHIP be integrated with Zephyr and SES
OpenThread is already integrated with Zephyr and distributed with it

Is Zephyr RTOS primarily for wireless applications or any embedded application?
Any embedded application, it's a general purpose RTOS

Whats the difference between Zephyr and standard RTOS?
If you are comparing Zephyr to the classical concept of an RTOS (the most well-known of which is FreeRTOS), the difference is that Zephyr is not just a kernel. It includes drivers, protocol stacks, filesystems and everything else you need to develop a firmware image
In addition to what carles said, you can read a bit more on why zephyr vs. other RTOSes here on LWN:
https://lwn.net/Articles/824029/ you may recognize the authors ;)

Are there any published books on zephyr
I'm not aware of any published books, but the online documentation is here:
https://docs.zephyrproject.org/latest/reference/kernel/scheduling/index.html
While we have pretty extensive online documentation, it is a gap that we don't have any books. I'm talking to one person that's thinking of writing one, but not really far enough along to predict a date. It's definitely a gap!

is Zephyr RTOS pre qualified for high integrity safety critical applications? for example in automotive ISO 26262 as an Safety Element out of Context?
There is a safety team that is working towards 61508 certification with LTS 2 next year.

Is there any performance (power mainly) impact using Zephyr compared to bare metal?
There should be no performance impact nor advantage of Zephyr vs bare metal. In both cases the CPU will go to idle whenever there is nothing for the core to do, and in both cases the peripherals need to be put to sleep when inactive. Zephyr is currently redesigning its power management subsystem, and that will be useful for some specific use cases.

Is there Zephyr automation with nRF Connect like in the Toolchain Manager?
There is no 'single-click' installer like the NCS toolchain manager in mainline or \vanilla\" zephyr, but we do package a Linux SDK with compiler builds and other host tools for all supported architectures, and I believe support for other platforms is in progress". Apart from that, you can follow the instructions in the getting started guide in the mainline zephyr project docs to get set up

will the sdk work with eclipse IDE ?
I'm not personally an eclipse user, but there is an eclipse plugin you may want to look into here: https://github.com/zephyrproject-rtos/eclipse-plugin

I haven't seen much discussion around update infrastructure. MCUMgr continues to make progress but any full-featured backends in discussion?
We support Updatehub and Hawkbit now, aside from MCUMgr. Zephyr as a project tries to remain neutral in that regard

Are there significant differences between vanilla Zephyr and Nordic bundled?
Nordic synchronizes our zephyr tree with upstream regularly in both directions (as we contribute upstream and merge into our downstream). there are some differences at the platform level which are mainly due to the way multiple application images such as the bootloader chain and main application are combined into a single hex file for flashing during development. this affects flash partition management. Obviously there are also many additional samples and extra Nordic specific documentation and board support downstream, along with a few device drivers and other hardware support items.

Zephrynetworkint support 4G o 5G ?
Indirectly. Zephyr has support for external modems, and those can be 4G or even 5G. Those external modems are usually connected via UART

@carles: which is great! But afaik they don't share a lot of common code, making it harder to swap out implementations. Something like the logging backend would be really neat to see.
The image manager is part of core Zephyr and all of those mechanisms use it (or should use it). Not sure what you mean about the logging backend?

What is latency resiliency?
Zephyr has a few features to reduce the latency of your application. When it comes to your ISRs (Interrup Service Routines), Zephyr has support for what we call Zero Latency IRQs (ZLIs) which ensure that your ISR runs at a higher priority than the kernel itself. We also have direct ISRs, which skip the preamble that is usually executed before the ISR to go directly into the ISR code. If you want low latency in code running in Thread mode, I recommend you take a look at meta-IRQs, which are similar to Linux's tasklets

Can we use the Zephyr on Github directly, or do you use a Nordic port of Zephyr. How does that work?
nRF Connect SDK (NCS) is the officially supported Zephyr based SDK for Nordic devices. You can, if you wish, run upstream Zephyr on nRF SoCs, but there's no official DevZone support for that, although several Nordic engineers are active members of that community as well. Some features in NCS are not present upstream.

hi from montreal. in the slide Native Execution on Posix compliant OS, you mention debugging/profiling your embedded application on a PC. How do you 'fake' what the real hardware inputs for that exercise?
We have support for simulating several hardware peripherals in our \native POSIX\" simulation builds. We support flash access via FuSe, displays, I2C, Bluetooth Low Energy, and we working on GPIO and SPI. All of those are purely based on simulations"

carles: ah sorry, I meant custom frontends which allows one to implement a different backends (like a custom service.) https://docs.zephyrproject.org/latest/reference/logging/index.html#custom-frontend. I should dig into the image manager more!
Right, MCUMgr for example supports at least 3 different transports/backends

Support for MQTT-SN?
No, not yet.

Is the visual studio code only available for linux? Or can we setup this nice development interface on Windows OS?
You're looking at VS Code on windows right now actually!

how to change IO configuration
Using devicetree and Kconfig

What board is he using for this example?
https://docs.zephyrproject.org/latest/boards/arm/reel_board/doc/index.html

Are there any infrastructure for firmware signing and secure boot?
Zephyr supports signing out of the box for the particular case of MCUBoot as a bootloader. It also supports integration with Trusted Firmware-M for Secure Boot. to complete my answer: the \west sign\" command supports MCUBoot but also third-party signing tools when properly configured"

Is your custom Python Kconfig something that is independant/usable outside of Zephyr?
Yes, https://github.com/ulfalizer/Kconfiglib

What IDE's are supported in Linux?
Eclipse and Visual Studio Code are fairly well supported

Are there plans for official support of other languages? (Primarily interested in Rust)
No current plans, but patches and contributions welcome

Is there a good reference for porting a board?   I have a board with an NRF52840 and I think I could use an existing Board to get started. Is there a wiki, etc for the process?
https://docs.zephyrproject.org/latest/guides/porting/board_porting.html