

Enterprise Working Group

First Meeting

Monday Aug 16, 2023

Attendees: Greg Wallace, Zoran Vujovic, Rafael Griman Canto, Johannes Kunde, Vedran Miletić, Ed Maste, Allan Jude, Joe Mingrone, Michael Osipov, Sebastian Muraru, Maxim Konovalov, Andrew Wafaa, Tomek CEDRO

Notes

- There was a discussion about FreeBSD for scientific computing and HPC that run these workloads. The consensus is that FreeBSD is great at the infrastructure layer, but lacks support for some key applications like SLURM
- Regarding Samba
 - Klara working on special Samba module to go into upstream samba to allow it to take advantage of new block reference tree ZFS feature.
 - To get upstream support for FreeBSD in Samba there was a comment “How do we get FreeBSD workers in the upstream Samba CI system?”
 - Comment from Maxim that Cloud Native is extremely important
 - Allan mentioned XC: <https://github.com/michael-yuji/xc>
 - Questions about how this relates to runj
- Comments from Zoran regarding Nvidia GPU drivers:
 - Lots of interest in Nvidia GPU drivers for AI and for HPC applications...like nvidia h100/a100 with nvlink, L40, L40s. What is the status?
- Add to gaps list
 - eBPF
 - Zero trust FreeBSD builds
 - Reproducible FreeBSD builds
 - SBOM for FreeBSD
- How well do SALT, Puppet, Ansible support FreeBSD
 - All three are used in production and work great.
 - SALT
 - Installing on FreeBSD:
<https://docs.saltproject.io/salt/install-guide/en/latest/topics/install-by-operating-system/freebsd.html>
 - Using SALT:
https://papers.freebsd.org/2019/bsdcn/hendryxparker-masters_and_minions_or_the_dream_of_bsd_automation/
 - Ansible
 - Using Ansible with BSD hosts:
https://docs.ansible.com/ansible/latest/os_guide/intro_bsd.html

- Quick start guide:
<https://blog.andreev.it/2021/02/ansible-quick-start-guide-for-freebsd-centos-and-ubuntu/>
-  Using Ansible Playbooks to automate deployment of various...
- Puppet

Second Meeting

Monday, Sep 4, 2023

Here is the link to the meeting recording: <https://youtu.be/rCRI57tdHns>

TL;DR This table summarizes the status of the priority projects. The rows in blue seem to be the ones we can execute against right away:

Project	Clear next step	Volunteers		
		Project Lead	User/Tester	Developer
Cloud Native / OCI runtime	Yes	Greg W	Yes	Yes
AD / DNS	Yes	Michael O	Yes	No
NVIDIA GPU	Yes	Greg W	Yes	No
Samba	Yes	Greg W	Yes	Yes
Definitive manager bhyve/VM/Jails	Yes	Chris Moerz	Yes	No
Zero Trust Build	Yes	FreeBSD Foundation	Yes	No
Ports Automation	No	FreeBSD Foundation	Yes	No
Enterprise CA	Yes	Greg W	Yes	Yes

Since this working group can likely make limited progress on Zero Trust Builds and Ports Automation, I suggest we move those into the parking lot and instead put Better certificate authority support in scope, as there seems to be a clear next step that can be addressed by the community.

We reviewed the results of the Feature and Infrastructure Priority and Difficulty survey, and began to plot next steps to work on the highest priority ones.

Third Meeting

Monday, Oct 2, 2023

Attendees

Greg Wallace, Ian Evans, Kyle Wilcox, Ed Maste, Chris Moerz, Jason Morris, Joe Mingrone, Brian Hodgins, Vedran Miletic, Pawel Dawidek, CeDeROM

Agenda

We talked through the status of the in-flight projects, specifically:

- Samba
 - We need to assess the patch situation and then contact Samba upstream. We have a couple folks who are working on this.
 - We also need to get FreeBSD officially supported in .NET and GH Actions - I am working on this
- OpenJDK
 - We opened a position to find a contractor for this today 🙌
- Cloud Native
 - Johannes made great progress on the PRD and I have invited a couple others to weigh in. Once it's complete, I will ask the group to review and approve.
 - Then we will move on to evaluating the current approaches against the PRD
 - Then on to developing
- NVIDIA
 - Excellent call today with someone at NVIDIA who wants to help us get FreeBSD support on Nvidia. Thanks Ian Evans at Metify for making this intro!!
- Bhyve manageability
 - Chris Moerz, a long time FreeBSD user and recent contributor and a project management leader, has stepped up to project manage this one. Thanks Chris!
 - He has updated the PRD and scheduled calls with bhyve users to collect input to further inform the PRD
 - Please reach out directly to Chris if you use bhyve and want to voice your thoughts on manageability
 - Expect a read out on status at the next meeting.

Third Meeting notes from the transcript

I shared out the agenda last week, and bear in mind we want to make sure that this is a really welcoming community for everybody who wants to participate, so keep keep the code of conduct in mind.

Charter: Why are we doing this? It's really to make sure that we can continue to support people who want to use FreeBSD as a general purpose server for enterprise workloads. But there's obviously overlap right? There's stuff that that have that has surfaced through this effort that is important, very broadly. One that comes to mind for me that are important beyond the general purpose server OS is the cloud native oci Runtime piece, and then the Nvidia gpu, and cuda,

Cloud Native/OCI

there's been a lot of strong interest in the Oci and Cloud native work and several folks are volunteering to work on that. I know johannes has made a start to the prd for that piece of work. So a lot of good stuff going on there.

Ian Evans: environment that uses a an oci framework and move it into Psd eventually. So we've already started to compile some of the requirements and some of the that we would need to see, at least from our application perspective. So we can start to maybe put some of that together, starting point and see if we can build, make maybe a framework or something.

Samba

I have a couple subtasks in here. So there's assess the patch landscape and I have a Google Doc in there with some notes. the other thing was to work on the github actions. So this was the thing that we were talking about a couple of minutes ago about the.net thing. So Github has become the default way that open source projects develop, and github actions has become if not the default, a leading way, that open source projects do their ci

so if you want to work upstream with anything like, say Samba or Python, or anything, really, that's using Github actions for their Ci, the only way to connect in an automated fashion, so that when there's a new version of that software, so in this case, Samba, it can automatically check to make sure that it's, you know, running well in in your software. So in our case, FreeBSD, you have to be supported by Github actions.

And as of today, FreeBSD is not. And the reason why we're not is because FreeBSD is not officially supported.net. So this is a problem.

we have been going after this pretty intensely. Somebody who I worked with quite closely when I was working in the node community and at the node foundation is now a director of product management at Github, not on Github actions, but I reached out to him, and he said he would be happy to send a message to his colleagues.

Deb, our executive director sent an email to the head of open source at Github, basically reiterating the same message.

I met with the Executive Director of the Open Source Initiative and was mentioning this challenge that we're currently facing. he introduced me to the Microsoft board member on OSI who in turn connected me with the head developer relations at .net.

And a person who I also used to work with at node in the node community, who is now like a head of developer relations at twilio works very closely with some people on the azure side. He connected me and we now have a great thread going with the people who can help us get the needed support in .NET and HitHub Actions.

OpenJDK

The Foundation just opened a position for a contractor to work in this. Check it out:
<https://freebsd.foundation.org/openjdk-contract/>

Joe: the goal is to get us to get official support for open jdk, on FreeBSD, I mean, initially, I think we'd focus on arm 64 and Aarch64.

Active Directory and DNS

Next step is to have a quick call, and I don't think that that's happened.

Greg is taking the action to followup on this one.

Nvidia

This is about FreeBSD drivers and cuda

We want to approach this similarly to the oci one, since this is a pretty big effort, I think we would want to look at what capabilities we think are really key.

And then start to think about how we, how we can approach it.

Ian Evans: Have you talked to Elad Ed Blatt and those guys over there at Nvidia?

he's got a little more responsibility across the entire organization for most of this stuff, including driver development. Including around the development of the Bsd stuff. And that's where we kind of started to work with them a little bit, but it was more around.

Ian Evans: The redfish enabled capabilities that they have on the Dpu. But he would probably be a really good person.

Vedran Miletic: if I may, I would also like to be included in this effort I have actually looked into Linux lotter based approach for running. Give me the drivers and cuda obviously available, but some parts are Linux specific, and then cuda is also Linux specific completely, and there are some hectic approaches. Maybe this would be a good first step. and I will also be interested in the deep use side if we can also eventually get that as well. That would be nice.

The reason why I ask is because I actually have a use case that I would like to run on previous. But I'm pretty much involved in development of the mostly CPU code, but also parts of the Gpu code. So this would be like a perfect test case to see whether we get like 94% of Linux performance will get 55% of Linux performance.

Chris Moerz: I'm wondering when we were talking about video drivers, are we talking implementing this natively, I suppose because when we're talking graphic stack as far as I remember there's the drm-kmod as a separate undertaking which may fuse with the LinuxKPI, which is a layer to basically bring in Linux code to speed up the development. I'm just wondering whether this is something worthwhile to investigate. Also for this, or whether we're actually focusing primarily on implementing something natively in FreeBSD.

Greg Wallace: anybody able to speak to that?

Ed Maste: my expectation is that virtually all of the work in this sort of area would be done using Linux Api. I mean, it is not it's not an emulation and there's not a lot of performance impact from using this Api. it's really just providing interfaces that whatever the software we want to port over requires and at the end of the day the requirements of whatever it is we're trying to port sort of already exist. And so, you know, it may well be that there's, you know, a slightly more sort of idiomatic way of doing something natively in freebsd or things of that nature. But practically speaking, you know, most of the time we're gonna end up having to implement the functionality, either missing functionality and freebsd, and then porting the driver to use that or just providing the interface via Linux api and so I yeah, I think it's the most likely basis is that all these sorts of things will be done.

Chris Moerz: Greg. I guess that's probably relevant in terms of when we're looking for developers today. Well, that just needs some kind of understanding in terms of graphic stack, graphics and drivers development. But also Linux Kpi. obviously, there's a couple of developers already working on their end team and it probably would lend itself to talk to these guys. But, as far as I remember, they're already pretty busy anyways.

Vedran Miletic: Yeah, if I may jump in. So if we are targeting cuda, then the only option is in video proprietary driver and not new law. And drm, related stuff from Linux kernel because CUDA doesn't support all the other other types of compute.

So the only that could do that inside of the driver, and not via some kind of layers of compatibility, or whatever whatever we wanna call them is in video guys that are actually already developing the driver and their who does support. And they're who the libraries, etc.,

etc. So no is really great to have for other use cases steam games whatever. In this case, we actually need the Nvidia driver. There is no way around.

Greg asked if One API is something to look at.

Vedran Miletic: So one Api you asked about, this is basically Intel at the moment. It is in. In theory, cross vendor. The problem is that it's very likely to suffer the same fate as Open Cl. The question in the end remains, how many applications choose to use all one Api instead of cuda, because many applications in the past actually used opengl in addition to cuda. And then everybody had video Gpus and use cool that have better performance than other Gpus via opengl. And you know in the end, basically open cl slid down. So it depends on you know, both the hardware from Intel and the chance to get the good performance on Nvidia. At least, let's say 95% could not 70% or or 50% or something.

Pawel Dawidek: maybe a general comment from me. like all those projects are pretty much focus on like chasing Linux. Are there anything like on the roadmap which could be like a selling point for FreeBSD? Something that could use to advocate FreeBSD as being innovative.

Greg Wallace: that's a great point, I mean. if I go back to the charter, this group, the Enterprise working group really came together in response to some specific feedback from a FreeBSD user who is hitting limits with FreeBSD as a general purpose enterprise server. So that's one use case. In that use case there are some limits that are a little bit more problematic: with FreeBSD than with Linux, but I think there's a lot of other use cases where FreeBSD is clearly not at all chasing Linux. Like for appliances for networking gear for security gear for where you need to be able to optimize for speed, performance and throughput FreeBSD really excels. the unambiguity of the Bsd license.

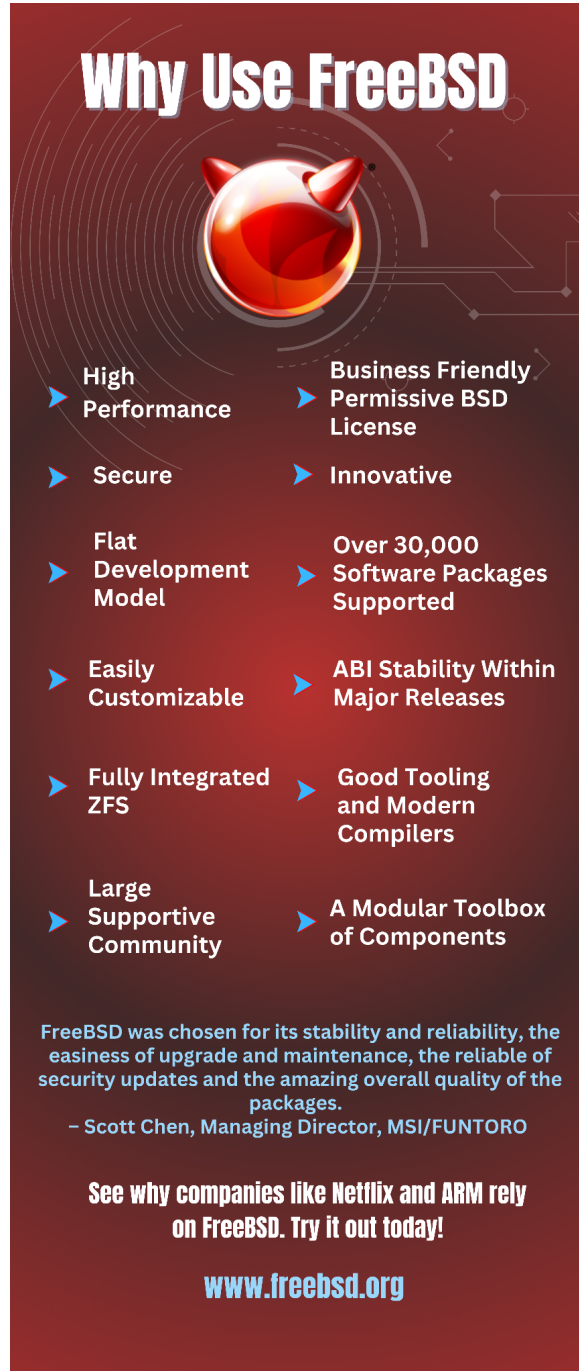
Ian Evans: And Greg, one thing I was gonna mention is kind of a fork in the road that I see in terms of some native capabilities that are really there's so much potential is around bhyve. just due to the fact that people are tired of paying the enterprise licensing on vmware boxes out there. It's not bad. I mean, there's some good Hypervisors, but that one alone. The the one thing that's a blocker on that for a lot of folks that I kind of talk to is the manageability of it. So if we look at like Church's project, you know where he's got some some pretty good wrappers, all shell based stuff, but that that's one that I can think of that really pushes a differentiation. And the Bsd realm, that's not necessarily, you know, chained to Linux or chain to some other OS, it's you know Joint? I know joint worked on that partially, but it it truly is a native technology in that sense that that, I think, really provides some some significant benefits as long as the manageability is in a better state.

Pawel Dawidek: These are all areas where we are chasing Linux. Can we also prioritize some of the areas where FreeBSD leads?

Greg Wallace: This specific WG formed around a set of limits that people detailed in trying to use FreeBSD as a general purpose enterprise server. That said, we are really working hard to make sure our marketing highlights all the areas where FreeBSD excels.

We've got a conference coming up and the Banner sign that our marketing team put together, calls out these things that FreeBSD really excels at.

[Here's that banner]



Why Use FreeBSD

- ▶ High Performance
- ▶ Secure
- ▶ Flat Development Model
- ▶ Easily Customizable
- ▶ Fully Integrated ZFS
- ▶ Large Supportive Community
- ▶ Business Friendly Permissive BSD License
- ▶ Innovative
- ▶ Over 30,000 Software Packages Supported
- ▶ ABI Stability Within Major Releases
- ▶ Good Tooling and Modern Compilers
- ▶ A Modular Toolbox of Components

FreeBSD was chosen for its stability and reliability, the easiness of upgrade and maintenance, the reliable of security updates and the amazing overall quality of the packages.

– Scott Chen, Managing Director, MSI/FUNTORO

See why companies like Netflix and ARM rely on FreeBSD. Try it out today!

www.freebsd.org

Vedran Miletic: just a quick comment to actually follow up after what time said. we actually evaluated the meat regarding virtual box and vmware and other appliances for virtualization. We

actually evaluate the FreeBSD, and there is TrueNAS core which has actually very good web interface for management of bhyve virtual machines. The only reason why we didn't take it is because it's for future is kind of uncertain right now, because has also started shipping Linux. and they look like more oriented towards shipping Linux in the future.

But the user interface from the users perspective from the admin's perspective looks really great, and it's really comparable to what proxmox offers, perhaps not as slick as you know, polished, etc. But it's almost there. So maybe that's something that we should also take into consideration to see if we could rip that out and then reuse it.

Pawel Dawidek: But they fully agree that, like we, we have this problem for a long time now, where we have some like good core technology. But the user experience is is really terrible. we had jails first, but then docker came along, and because how easy it was to use it simply revolutionized the marketing.

I think the poor user experience was a big reason why we lost out to docker. it wasn't possible to install FreeBSD on Zfs from the _____. So yeah, this is like problem we have for a long time. we don't really care about like unexperienced users and how to improve user experience for them.

Greg Wallace; Even though I'm new, we've heard this quite a bit. The technology needs to be accessible to people. It can't only be valuable to people who've already been using it for many of years. It has to be something that is approachable and accessible to people who are newer.

Vedran Miletic: I actually want to add something about zfs, they don't, wanna, you know, jump on everything. But there is a success story. So I don't wanna end it on a you know. Like a pessimistic note actually perfect since version 12 or so. That's when I gave it another shot. I use it as a third party modules for much longer than that. But zfs, in installer you use version 12 or later you can get the root on Zfs from the basic installer by just selecting the drives that you want to use. That's impossible with Linux. If you want any kind of rate configuration. You will be configuring stuff left and right, and most installers don't even support zfs. that's true for ubuntu, which actually ship zfs on the desktop for a while, and that's true for Debian at least, but probably others as well. So zfs support is actually a point where, whether you choose to use the default FreeBSD Use root on zfs or whether you choose to go for the appliance to get extra storage. Zfs, we have actually done very, very good work, and not to mention that when you do such exotic stuff, let's say 6 drives into raid, 5 zfs. Root on zfs, and you pull out a drive the bootloader, Another drive takes over. So everything is set up exactly at some of the do it, you know, manually, without actually requiring any manual knowledge or intervention, a great success story where somebody really thought what the user would like to see and then make made it happen. Even the upgrade notes actually mentioned If you use zfs with Ufi, you need to update both loads. So each of the both partitions like this, and then you get the the commands to do it. So zfs, support and user experience, even for the new users that don't know how to manage the Fs is actually very well done.

Fourth Meeting

Monday, Oct 27, 2023

Attendees

Greg Wallace, Ed Maste, Chris Moerz, Michael Dexter, Joe Mingrone, Antranig Vartanian, Dave Cottlehuber, Michael Osipov, Antranig Vartanian

Agenda:

- Housekeeping
- Big picture status
 - bhyve manageability project update - Chris Moerz
- jailer/jailerctl demo and a bhyve supervision demo - Antranig Vartanian
- AD, OpenJDK updates - Joe Mingrone
- OCI FreeBSD extension, .NET/GH Actions, AI support update - Greg Wallace

Slides  Enterprise_WG_Meeting_4

Call transcription:

<https://drive.google.com/file/d/14rBcDIw4KQ8Ex1ioxlfEOJnlCkaHm98s/view?usp=sharing>

Recording <https://youtu.be/RgF53gJjiNg>

Dave Cottlehuber comments on Jails management

a lot of previous work development tends to be done without much community visibility. And then we have someone bringing something really awesome to the table in quite a finished state.

I'm hoping to get out of our with regular jails working group calls. It's something that has wider community input that obviously takes longer but then, hopefully, something that we could have in base that addresses to me one of the fundamental issues with Jails that we have, which is that there are many, many tools to choose from, all supported by one person in their free time, and to end up with something that is well integrated from use cases and able to support it and base. This would be a great outcome for me.

What we're proposing is

1. to extend the kernel to support some arbitrary per jail metadata and some basic state awareness. when I say state awareness, I mean this: it should be possible from outside to understand where the shell is running stopped or in progress, it being set up. That's probably about it. This lines up very nicely to the oci model for the state machine. And we should be able to use pretty much exactly the same thing we have.
2. The second piece is to add a userland demon that exposes this state data over some sort of socket Api and provide the ability to run hooks when these states change from user perspective. The background to this is, if you looked at Docker on Linux, this is a lot of the sort of its functionality: it provides this overview of all the containers that are running and is able to make broad changes based on state of that from the outside as a manager rather than from the inside, from the perspective of the jail.
3. The third piece is to provide a package like tool that can fetch, build, and publish what I call a "jails images", but they effectively travel with minimal metadata. And again, if that is something that we can line up with Oci spec in so much the data.
4. And then fourth, which is perhaps more aspirational, is to extend the kernel to support non root jails. When I say non root jails, I mean the ability for normal non root user to create a jail. It doesn't have to have room inside the jail, but they should be able to do that. And again, this is another piece of functionality that ____ more complicated workflows, easy to produce on previous thing with a much lower security. Yeah. Security We're looking for here.

There's a long list of unfettered data which I I've seen through to Chris and Greg already, and maybe can attach it to the meeting minutes.

FreeBSD Jail Desired Features

But the main thing out of this process, that I'm looking for, is one consensus on what we're building to seek developers and funding for that as well. Those are kind of the 2 next pieces. We've got some people doing some of the work and the next pieces to find a way to make this.

Chris Moerz, bhyve (Jails) manageability workstream

(Slides 5 through 23)

Joe Mingrone

OpenJDK update

Antranig Vartanian, jailer/jailerctl demo and a bhyve supervision demo

