AArch64: Bringing a New FreeBSD Architecture to Tier-1

BY ED MASTE

FreeBSD descended from 386BSD and began with support for a single CPU architecture, the Intel 80386. Support for a second architecture, DEC Alpha, arrived in FreeBSD 3.2 and support for 64 bit x86 (amd64) came next. The concept of support tiers wasn't yet fully cemented, but amd64 was promoted to Tier-1 status in 2003. The 64-bit Arm architecture AArch64, also known as arm64, obtained Tier-1 status in 2021. We'll explore what that means and how we got here.

Bringing up a new Tier-1 architecture in FreeBSD is a challenging task, and it requires significant effort to ensure that the architecture is fully supported, stable, performant, and compatible with the existing FreeBSD ecosystem.

The FreeBSD Foundation saw potential in 64-bit Arm, and also learned of other entities with an interest in a FreeBSD port to the platform.

Tier-1 Status

The FreeBSD project website [https://docs.freebsd.org/en/articles/committers-guide/#archs] documents the three support tiers. Tier-1 references fully-supported architectures, Tier-2 is developmental or niche architectures, and Tier-3 are experimental architectures.

The documented Tier-1 status refers primarily to the guarantees the FreeBSD project makes with respect to the architecture, including generating release artifacts, providing prebuilt packages, support by the security team, and backwards compatibility goals across updates. Tier-1 also implies that the platform is actively maintained, regularly tested, and receives timely bug fixes and security updates. A Tier-1 platform is expected to be fully integrated into the FreeBSD build system so that all components of the tool chain are functional and developers can easily build, install and maintain the operating system on that platform.

Tier-1 status also covers some implicit characteristics, such as hardware availability. FreeBSD doesn't explicitly require Tier-1 platforms to be widely available or popular, but in practice, Tier-1 status requires that a variety of hardware platforms exist and are available with a reasonable cost. This is because FreeBSD relies on a combination of community support and vendor contributions to maintain and improve its support for different hardware platforms and to build and test third-party software for the architecture.

Tier-1 platforms are also expected to be self-hosting—that is, it is possible to build a new version of the kernel, C runtime, userland tools, and the rest of the base system on FreeBSD on that platform.

Platform Genesis

Like several other platforms, FreeBSD/arm64 began with the interest of a motivated developer. Andrew Turner is a longtime FreeBSD/arm developer who started looking at Arm's AArch64 architecture shortly after it was announced. The FreeBSD Foundation saw potential in 64-bit Arm, and also learned of other entities with an interest in a FreeBSD port to the platform. The Foundation formed a project to coordinate and sponsor both Andrew Turner and engineering firm Semihalf, with the support of Arm, and CPU vendor Cavium.

The earliest commit in the FreeBSD that references arm64 added build infrastructure for the kernel-toolchain build target. As the name suggests, this target builds the tool chain (the compiler, linker, and so on) that is then used to compile, link and convert a kernel. Clang was part of the FreeBSD base system at the time this work was done, so compiler support was fairly straightforward. However, at the time, FreeBSD still included an older version of the GNU ld linker, which predated support for AArch64. Thus, the early build support relied on having the aarch64-binutils port or package installed, and it used the provided linker automatically.

First kernel change for arm64:

```
commit 412042e2aeb666395b3996808aff3a8e2273438f
Author: Andrew Turner <andrew@FreeBSD.org>
Date:   Mon Mar 23 11:54:56 2015 +0000

    Add the start of the arm64 machine headers. This is the subset
    needed to start getting userland libraries building.

    Reviewed by:  imp
    Sponsored by:     The FreeBSD Foundation
```

After several years of development, FreeBSD had a basic, but functional, self-hosted FreeBSD/arm64 port, with some ports and packages available. A lot of development effort, debugging, performance tuning, documentation, and other work remained, but FreeBSD was on the path to adding another architecture to the supported list. FreeBSD 11.0 became the first release to include arm64 support and installable artifacts, as a Tier-2 platform.

Tool Chain

Beginning of arm64/AArch64 support:

```
commit 8daa81674ed800f568b87f5e4b8881d028c92aea
Author: Andrew Turner <andrew@FreeBSD.org>
Date:   Thu Mar 19 13:53:47 2015 +0000

    Start to import support for the AArch64 architecture from ARM. This
    change only adds support for kernel-toolchain, however it is expected
    further changes to add kernel and userland support will be committed
    as they are reviewed.

    As our copy of binutils is too old the devel/aarch64-binutils port needs
    to be installed to pull in a linker.

    To build either TARGET needs to be set to arm64, or TARGET_ARCH
    set to aarch64. The latter is set so uname -p will return aarch64 as
    existing third party software expects this.

    Differential Revision:  https://reviews.freebsd.org/D2005
    Relnotes:       Yes
    Sponsored by:       The FreeBSD Foundation
```

One of the first requirements for a Tier-1 platform is to have a fully-supported, integrated tool chain. Clang is the primary compiler used in building FreeBSD and it has received significant and ongoing AArch64 development effort from several large companies. Thus, compiler support was quite good for the entire duration of platform bring-up.

Other tool chain components, like the linker, debugger, and miscellaneous binary utilities needed more work. As initial FreeBSD/arm64 porting effort was nearing completion, FreeBSD still used the GNU binutils linker ("BFD linker") and had not updated the version of the linker for some time due to licensing concerns. As a result, the linker included in the base system did not support AArch64 and initial support depended on having a binutils port or package installed. We made this as convenient as possible for end users, but it did not meet the requirements for a Tier-1 architecture.

Fortunately, rapid progress was also being made on LLVM's LLD linker within the LLVM community. LLD offered the potential for much faster link times, facilitated optimizations not available with the BFD linker, and wider architecture support. At the end of 2016, we were able to switch to using LLD as the system linker for FreeBSD/arm64. In fact, it was the first FreeBSD architecture to do so.

FreeBSD uses the ELF Tool Chain project for miscellaneous binary utilities such as `strings` or `strip`. These have some machine-dependent functionality (such as lists of relocation types), but the effort required to add arm64 was relatively small.

The final tool chain component that required significant development effort was LLDB, the LLVM family debugger. Fortunately development work was being done to support other operating systems, and it required only incremental effort to add FreeBSD.

We were able to merge much of this tool chain work into the FreeBSD 11 stable branch, and FreeBSD 11.1 was the first release to avoid workarounds and include a functional linker.

Ports and Packages Collection

FreeBSD provides over 30,000 third-party software packages in its ports collection, and many of these have architecture-dependent characteristics. Machine-dependent infrastructure (e.g., controls for a given port to opt-in or opt-out of building on a given architecture) are fundamental parts of the ports tree. FreeBSD/arm64 was available as a Tier-2 architecture and FreeBSD community members experimented and discovered ports that failed to build. These were either fixed, or excluded from building on aarch64 if appropriate. Mark Linimon was one of the key developers who took on this work.

The goal of bringing FreeBSD/arm64 to Tier-1 brought with it some additional requirements of the ports tree. The ports collection does not have an official hierarchy or tier categorization of ports, but there are some ports that are critical. This includes ports that provide tool chain components or other dependencies required to build large ports of the full collection. We had to make sure that these were available and consistently built for FreeBSD/arm64.

Bringing a new architecture to Tier-1 status requires the support and agreement of several teams within the FreeBSD project.

We also need to build package sets in a timely manner for Tier-1 architectures, which requires capable server hardware. The FreeBSD Foundation purchased servers from Ampere Computing, and the project received additional servers donated by Ampere. This hardware allows the arm64 package sets to be built on the same weekly cadence as x86

Support from FreeBSD Teams

Bringing a new architecture to Tier-1 status requires the support and agreement of several teams within the FreeBSD project. This includes the ports management and package management teams as mentioned above, along with the security team, release engineering team, and the core team.

The Release Engineering team is responsible for building and testing release artifacts including ISO and USB memory stick images, as well as cloud computing targets. These artifacts can be cross-built from other architectures, so arm64 build hosts are not absolutely required by the Release Engineering team, but test and QA hardware is needed.

To be Tier-1, an architecture requires the security team to provide source updates for security issues and errata as well as binary updates via freebsd-update.

Finally, the core team's support is necessary to coordinate with the other teams, with the community, and make the official declaration that the platform is officially Tier-1

Hardware Ecosystem

An implicit requirement for a platform to be Tier-1 is the availability of suitable hardware, as alluded to earlier. Hardware is needed at many different price/performance combinations:

- high-end servers to build packages,
- mid-range, sever-class hardware for developer workstations, remote access for porting and testing, and so on,
- low-end embedded style platforms for ubiquitous testing and developer use,
- cloud resources at various levels for development, testing, and production.

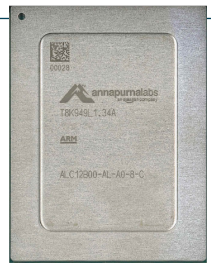
AArch64 started with some notable gaps in the available hardware, in particular, related to mid-range (and mid-price) platforms for developer and porting efforts. There were very few options in the mid-late 2010s. The SoftIron OverDrive 1000 was a well-priced, capable system in a convenient developer form factor based on the AMD A1100 processor. Unfortunately, both the A1100 and the OverDrive 1000 were discontinued not long after being introduced.

Hardware availability continues to improve with platforms like the Raspberry Pi 4 and Pine A64-LTS at the lower end, Apple devices and the Microsoft AArch64 developer platform in the middle, and high-end Ampere Altra-based server systems. AArch64 virtual machines are also offered by major cloud vendors, using either Ampere platforms or a bespoke CPU design (AWS' Graviton).

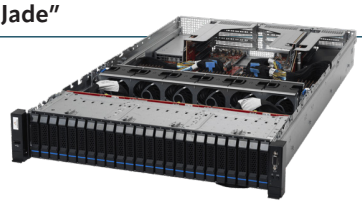
Bringing FreeBSD/arm64 to Tier-1 status required a significant investment of time and resources. The 64-bit Arm ecosystem has captured a significant portion of the server market, with no signs of slowing down. FreeBSD will benefit from tapping into this market with this Tier 1 platform.

ED MASTE is the Senior Director of Technology for the FreeBSD Foundation, managing the Foundation's technology roadmap, development team and sponsored projects. He is also a member of the elected FreeBSD Core Team for the current term. Aside from FreeBSD, he has contributed to a number of other open-source projects, including LLVM, ELF Tool Chain, QEMU, and Open vSwitch. He lives in Kitchener-Waterloo, Canada, with his with his wife Anna and children.

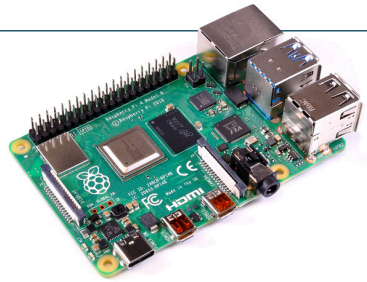
AWS Graviton



Ampere "Mount Jade"



Raspberry Pi 4



Microsoft Arm Developer Kit

Pine A64 LTS