

FreeBSD/sparc64 status

Marius Strobl
marius@FreeBSD.org



FreeBSD Developer Summit
Meeting Plaza
Maarsse, The Netherlands
October 6 – 8, 2011

Userbase

- ▶ Hard numbers are unknown, but based on private and public e-mail the FreeBSD/sparc64 user community appears to be vivid.
- ▶ Breakages usually are reported within days, even for head.
- ▶ Back when `freebsd-update(8)` did not support amd64, the number of attempts to use it on sparc64 were close to those of amd64 according to `cperciva@`.
- ▶ After Oracle changed the Solaris licensing, the interest in FreeBSD/sparc64 increased.
- ▶ Apparently it is also used for commercial services.
- ▶ Unfortunately, most are ordinary users and not developers, i.e. non-contributing.



Usability

- ▶ FreeBSD/sparc64 currently runs on machines based on Sun/Oracle UltraSPARC-I up to UltraSPARC-IV+ and Fujitsu SPARC64 V CPUs with some white spots due to them re-inventing the wheel with nearly every machine model.
- ▶ Stability is (mostly) en-par with x86, like on the other non-x86 architectures not all features (DTrace, `freebsd-update(8)`, `hwpmc(4)`, `minidumps`, `superpages` etc) are available though.
- ▶ Catches a lot of issues (64-bit, alignment, endianness, toolchain etc) that equally affect the other non-x86 architectures.
- ▶ Thanks to flo@, beat@, linimon@ and others the availability and quality of ports and packages significantly increased over time, although the quality of third-party software on 64-bit big-endian was on its top when Linux/PPC64 was popular.



FreeBSD

- ▶ CPU`TYPE` support added ("ultrasparc", "ultrasparc3" and "v9")
- ▶ Support for a feature of UltraSPARC-III+ and later obsoleting the needing for locking the kernel TSB into the dTLB and thus allowing to use the whole 64-bit kernel address space was added (before: theoretical maximum 16GB).
- ▶ As a follow-up KVA was increased to not be limited by `VM_KMEM_SIZE_MAX` and `VM_KMEM_SIZE_SCALE` was decreased to 1 allowing kernel to use more memory (analogous to amd64, f.e. for ZFS).
- ▶ The `iommu(4)` driver has been changed to take advantage of the streaming buffers (after auditing all device drivers in `GENERIC` for `bus_dmamap_sync(9)` issues and fixing them as necessary).



2011 continued

- ▶ Following the update of the binutils to 2.17.50 (first in-tree version to support GNUTLS on SPARC), support for TLS relocations was added to rtdl(1) and TLS-support was enabled in the base GCC and malloc(3) for sparc64.
- ▶ schizo(4) has been updated to also support XMITS host-PCI-X bridges and a workaround for Cassini/Skyhawk combinations has been added (may solve the crashes seen when using the the on-board Cassini NICs of Sun Fire V480 equipped with defective centerplanes).
- ▶ The largeSMP project caused a lot of fallout in the sparc64 C and assembler code but as a result MAXCPU now defaults to 64 (more possible but currently not required).
- ▶ The code responsible for flushing user mappings from the TLBs of UltraSPARC-III/SPARC64 V and later was re-written to scale beyond 8 cores (tested w/ 16-way V890).



2011 continued ...

- ▶ libthread_db support was added, unbreaking kgdb(1) on sparc64.
- ▶ The implementations of atomic operations was optimized performance wise.
- ▶ MD bus drivers were update for NEW_PCIB.
- ▶ MD use of sched_lock was replaced with atomic operations.
- ▶ Support for SCHED_ULE was added to cpu_switch().



Future and obstacles

- ▶ Fix the userland mutex code. After running pho@'s stress test for some hours on a 16-way machine, none of the threads can obtain the lock any more. Apparently this is due to a race condition and/or missing memory barriers. Help from people knowing how this code actually works is highly appreciated.
- ▶ Add support for the current line of SPARC64 VI/VII based Fujitsu/Oracle SPARC Enterprise Mx000 Servers. Most of the required bits (CPU, Oberon host-to-PCI-Express bridge etc) are largely already there. Problem: no hardware
- ▶ Re-add support for sun4v machines using a shared/common FreeBSD/sparc64 code base (later work there was already done with this in mind) instead of a separate FreeBSD/sun4v. Problems: FreeBSD/sun4v never was mature and had lots of h0h0magic, sun4v has fewer trap levels, generally a lot of work, no hardware



Future and obstacles continued

- ▶ Clang so far does not support 64-bit SPARC v9, the current SPARC back-end has no active maintainer and generally still seems to be in its infancy (also: libcompiler_rt is broken by design on SPARC) – apparently the only option is to keep GCC as the system compiler for FreeBSD/sparc64 for the time being.
- ▶ It would be nice if developers would write "machine-independent" bits with non-x86 in mind. For example it took three people and a lot of work in several attempts to get at at least the initiator part of mpt(4) endian-clean. Then mps(4) came along again only written with little-endian architectures in mind (unlike its mpt2sas(4) rival) ...



Thank you for your attention!
Questions?

